EDF R&D



DEPARTEMENT LABORATOIRE NATIONAL D'HYDRAULIQUE ET ENVIRONNEMENT
GROUPE HYDRAULIQUE A SURFACE LIBRE ET EXPLOITATION

6, QUAI WATIER
F-78401 CHATOU CEDEX

TEL : 33 1 30 87 72 52
FAX : 33 1 30 87 80 86

Janvier 2007

VIOLEAU D., ISSA R.

Guide du logiciel SPARTACUS-2D V1P2 : calcul lagrangien d'écoulements bidimensionnels laminaires et turbulents par la méthode SPH

HP-75/04/046/A

Documents associés : Voir aussi : -
Annule et remplace : -
Annexe de : -

Résumé : Le logiciel SPARTACUS-2D a pour vocation de prédire une catégorie d'écoulements « exotiques », c'est-à-dire difficiles à aborder par des méthodes numériques classiques comme les éléments finis. C'est le cas, par exemple, des écoulements présentant une ou plusieurs surfaces libres complexes, éventuellement associées à des zones en charge. SPARTACUS-2D est un code bidimensionnel fondé sur la méthode numérique nommée Smoothed Particle Hydrodynamics (SPH), reposant sur un formalisme lagrangien pour la modélisation des écoulements laminaires ou turbulents quasi-incompressibles. L'acronyme SPARTACUS signifie « Smoothed PARTicle hydrodynamics for ACcUrate flow Simulation ». Le code, dans sa version 1.2 et muni de pré- et post-processeurs adaptés, est ici documenté pour le développeur et l'utilisateur. On trouvera dans ce document trois parties : notice théorique, descriptif informatique et mode d'emploi, enfin dossier de validation.

EDF R&D



DEPARTMENT NATIONAL HYDRAULICS AND ENVIRONMENT LABORATORY
GROUP FREE SURFACE FLOWS AND EXPLOITATION

6, QUAI WATIER
F-78401 CHATOU CEDEX

TEL : 33 1 30 87 72 52
FAX : 33 1 30 87 80 86

January 2007

VIOLEAU D., ISSA R.

Guide for the SPARTACUS-2D V1P2 code : Lagrangian modelling of two-dimensional laminar and turbulent flows with SPH method

HP-75/04/046/A

Related Documents : See also: -
Replace: -
Appendix to: -

Abstract: The SPARTACUS-2D code is devoted to the simulation of various kinds of « exotic » flows, i.e. uneasy to predict with traditional numerical methods like finite elements. Such difficulties are met when modelling flows involving one or several complicated free surfaces on one hand, and confined areas on the other. SPARTACUS-2D is a two-dimensional code based on the so-called Smoothed Particle Hydrodynamics (SPH) method, written in a Lagrangian formalism for the modelling of laminar and turbulent weakly compressible flows. The acronym SPARTACUS means « Smoothed PARTicle hydrodynamics for ACcUrate flow Simulation ». This code, herein in its 1.2 release and associated with appropriate pre- and post-processors, is documented in the present report. The reader will find three parts, namely: a principle note, a developer's and user's guide, eventually a validation document.

Auteur(s)	VIOLEAU D., ISSA R.
Code Action	E961/P00260/ELOT008
Classement Interne	Zone libre

Type de rapport	Guide logiciel
Nombre de pages	171
Orientation dans le fonds documentaire	☒ EDF DOC (accès à tous les agents EDF) ☐ R&D DOC (accès aux seuls agents R&D) ☐ CONFIDENTIEL (accès réservé à la hiérarchie de l'entité émettrice)
Mots-clés	SPH, CFD, lagrangien, logiciel, guide utilisateur, validation

Indice	Auteur	Vérificateur	Approbateur
A	VIOLEAU D. signé le 09/01/2007 ISSA R. signé le 09/01/2007	HERVOUET J.-M. signé le 15/01/2007	DESROCHES E. signé le 20/01/2007 ☑ Autorise l'exploitation de la version électronique de cette note* pour alimenter les fonds documentaires de Galaxie. * *sauf pour les notes confidentielles.*

	Destinataire	Dept	Nb		Destinataire	Dept	Nb
@	Fonds-documentaire	AGIR/CIVAP	1		Issa R.	LNHE	2
@	Chabard J.-P.	EDF R&D		@	Razafindrakoto E.	LNHE	
@	Perotin L.	LNHE			Buvat C.	LNHE	2
	Fournier J.-C.	LNHE	1	@	Potapov S.	AMA	
	Hervouet J.-M.	LNHE	1				
	Laurence D.	MFTT	2		Lang P.	SOGREAH	3
	Benoit M.	LNHE	1		Rogers B.	Univ. Manch.	2
	Violeau D.	LNHE	5				

Pré diffusion aux destinataires signalés par *	Diffusion : P pour pages de garde et contrôle, S pour pages de garde, de contrôle et de synthèse @ pour version électronique

Guide du logiciel SPARTACUS-2D V1P2 : calcul lagrangien d'écoulements bidimensionnels laminaires et turbulents par la méthode SPH

SYNTHESE

Le logiciel SPARTACUS-2D a pour vocation de prédire une catégorie d'écoulements « exotiques », c'est-à-dire difficiles à aborder par des méthodes numériques classiques comme les éléments finis. C'est le cas, par exemple, des écoulements présentant une ou plusieurs surfaces libres complexes, éventuellement associées à des zones en charge. SPARTACUS-2D est un code bidimensionnel fondé sur la méthode numérique intitulée Smoothed Particle Hydrodynamics (SPH), reposant sur un formalisme lagrangien pour la modélisation des écoulements laminaires ou turbulents quasi-incompressibles. L'acronyme SPARTACUS signifie « Smoothed PARTicle hydrodynamics for ACcUrate flow Simulation ». Le code, dans sa version 1.2 et muni de pré- et post-processeurs adaptés, est ici documenté pour le développeur et l'utilisateur. On trouvera dans ce document trois parties : notice théorique, descriptif informatique et mode d'emploi, enfin dossier de validation.

Guide for the SPARTACUS-2D V1P2 code : Lagrangian modelling of two-dimensional laminar and turbulent flows with SPH method

EXECUTIVE SUMMARY

The SPARTACUS-2D code is devoted to the simulation of various kinds of « exotic » flows, i.e. uneasy to predict with traditional numerical methods like finite elements. Such difficulties are met when modelling flows involving one or several complicated free surfaces on one hand, and confined areas on the other. SPARTACUS-2D is a two-dimensional code based on the so-called Smoothed Particle Hydrodynamics (SPH) method, written in a Lagrangian formalism for the modelling of laminar and turbulent weakly compressible flows. The acronym SPARTACUS means « Smoothed PARTicle hydrodynamics for ACcUrate flow Simulation ». This code, herein in its 1.2 release and associated with appropriate pre- and post-processors, is documented in the present report. The reader will find three parts, namely: a principle note, a developer's and user's guide, eventually a validation document.

TABLE DES MATIERES

Répertoire des modifications du document

Référence	Désignation des modifications	Observations

INTRODUCTION

Le logiciel SPARTACUS-2D a pour vocation de traiter une catégorie d'écoulements « exotiques », c'est-à-dire difficiles à aborder par des méthodes numériques classiques comme les éléments finis. C'est le cas, par exemple, des écoulements présentant une ou plusieurs surfaces libres complexes, éventuellement associées à des zones en charge. De tels écoulements sont assez fréquents ; on peut citer le cas des évacuateurs de crue, les écoulements dans les ouvrages de prise et rejet des CNPE, ou encore le déferlement des vagues sur un ouvrage d'eau côtier ou sur une pile d'éolienne offshore, etc.

SPARTACUS-2D est un code bidimensionnel fondé sur la méthode numérique intitulée *Smoothed Particle Hydrodynamics* (SPH), reposant sur un formalisme lagrangien (suivi de particules) appliquée à la modélisation des écoulements laminaires ou turbulents quasi-incompressibles. Récemment apparue (les premières publications datent de la fin des années 70), la méthode SPH est cependant relativement éprouvée dans le domaine de la mécanique des fluides. Elle est en outre d'une grande simplicité, tant pour le programmeur que pour l'utilisateur, et permet d'aborder numériquement une vaste catégorie de phénomènes, notamment la turbulence depuis les développements effectués par le LNHE ces dernières années.

L'acronyme SPARTACUS signifie « Smoothed PARTicle hydrodynamics for ACcUrate flow Simulation ». Le code, dans sa version 1.2 et muni de pré- et post-processeurs adaptés, est ici documenté pour le développeur et l'utilisateur. On trouvera dans ce document quatre parties : notice théorique, guide de programmation, notice d'utilisation et dossier de validation, réunis en un seul volume pour plus de commodité.

1. PREMIERE PARTIE : FONDEMENTS THEORIQUES

1.1 PRINCIPES DE LA METHODE NUMERIQUE SPH

La première partie de ce document est une notice théorique. Nous y présentons les fondements de la méthode numérique SPH, ainsi que les équations considérées par SPARTACUS-2D et la méthodologie retenue pour leur résolution. Nous n'exposerons pas les bases théoriques en détails et de manière exhaustive, préférant nous diriger d'emblée vers les hypothèses adoptées dans le cadre de la modélisation des écoulements quasi-incompressibles, et plus précisément vers les choix qui ont présidé à la construction du logiciel. On notera que les options retenues ici correspondent à un modèle bidimensionnel, le cas tridimensionnel ne présentant toutefois pas de différences majeures ; pour plus de détails, on se reportera à Monaghan (1992) et Morris *et al.* (1997), ainsi qu'à Issa (2005).

1.1.1 Formalisme mathématique

1.1.1.1 Interpolation

SPH est une méthode complètement lagrangienne : les quantités physiques sont attachées à des particules, et non à des points fixes de l'espace. Elle ne nécessite aucun maillage pour l'interpolation des champs, pas plus que pour l'écriture des opérateurs différentiels.

Dans l'espace physique, le fluide est représenté par un ensemble de N entités macroscopiques de fluide de volumes τ_a, centrés sur autant de points de l'espace Euclidien indexés par des lettres latines (a ou b). Chacun de ces points occupe une position notée par le rayon vecteur $\underline{r}_a$ (voir **figure 1**). Par abus de langage, nous appellerons « particule » ces points a, et nous affecterons à chacune une masse m_a, *constante* au cours du temps. Chaque particule a possède également une densité ρ_a, une pression p_a et un vecteur vitesse $\underline{u}_a$. Nous verrons plus loin qu'il est parfois nécessaire de leur affecter d'autres grandeurs.

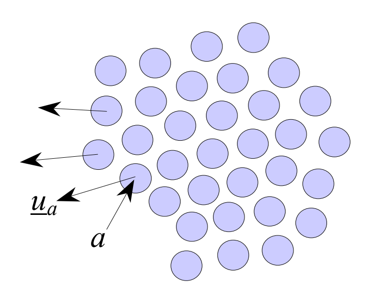


Figure 1 – La discrétisation fluide par la méthode SPH : un ensemble de particules.

D'une manière générale, une quantité physique A – champ scalaire, vectoriel ou tensoriel – mesurée au point occupé par la particule a sera notée A_a. Il faut entendre les valeurs des pressions et des vitesses comme des moyennes statistiques sur toutes les entités mésoscopiques de fluide (au sens de la Mécanique des Milieux Continus) que renferme le volume τ_a ; il en sera de même de la définition de toute grandeur thermodynamique ou mécanique intensive. Les grandeurs extensives, en revanche, sont entendues comme de simples sommations sur τ_a ; c'est notamment le cas de la masse[1].

Les densités des particules étant individuelles, modéliser un écoulement à plusieurs phases ne posera aucun problème. Dans ce cas, il faut en toute rigueur définir une viscosité pour chaque particule, ce

[1] Dans la suite de ce document, compte tenu du fait qu'on est en dimension 2, on gardera à l'esprit le fait que les « volumes » τ_a se comptent en mètres carrés, tandis que les densités ρ_a se comptent en kilogrammes par mètres carrés (densités surfaciques). Cette subtilité n'a cependant pas d'incidence sur les équations, qui peuvent être entendues comme « tridimensionnelles mais invariantes selon la direction perpendiculaire au plan de travail ».

que nous ne ferons pas dans ce chapitre, par souci de simplicité (la plupart du temps, on considère seulement un ou deux fluides). L'introduction de la modélisation de la turbulence (**§ 1.1.3**) comblera cette lacune.

Au fondement de SPH, se trouve la méthode d'interpolation suivante. La valeur d'un champ A suffisamment régulier, en un point donné $\underline{r}$, est tout d'abord écrite comme le produit de convolution de ce champ avec la distribution δ de Dirac (formulation mathématique exacte) :

$$A(\underline{r}) = \int_\Omega A(\underline{r}')\delta(\underline{r} - \underline{r}')dr' \tag{1}$$

l'intégrale étant étendue à l'ensemble du domaine Ω contenant le fluide et ses bords[2]. Pour des raisons numériques, la distribution de Dirac est ensuite approchée par un noyau $w_h(\underline{r} - \underline{r}')$[3]. L'intégrale de cette fonction est égale à l'unité (condition de normalisation) :

$$\int_\Omega w_h(\underline{r})dr = 1 \tag{2}$$

Le noyau est une fonction régulière qui « ressemble » à δ (ou plus précisément à une fonction Gaussienne) mais possède généralement un support compact, connexe et d'intérieur non vide. La largeur de ce support, noté ici h_t, est proportionnelle à un paramètre h nommé « longueur d'atténuation » (« *smoothing length* »). Dans SPARTACUS-2D, il est constant et proportionnel à l'écart initial moyen entre deux particules voisines, noté δr (et assimilé au « diamètre » d'une particule). Le rapport $\alpha_h = h / \delta r$, choisi par l'utilisateur, est généralement de l'ordre de 1,2 à 1,6.

Compte tenu de ce qui précède, on montre que

$$A(\underline{r}) = \int_\Omega A(\underline{r}')w_h(\underline{r} - \underline{r}')dr' + o(h) \tag{3}$$

Le noyau tend donc vers la distribution de Dirac lorsque h tend vers 0, c'est-à-dire lorsque la discrétisation du fluide sous forme de particules est de plus en plus fine. Dans SPARTACUS-2D, le noyau est sphérique, c'est-à-dire qu'il ne dépend que de la distance entre les points $\underline{r}$ et $\underline{r}'$. Ainsi, chaque noyau se présente sous la forme générale suivante :

$$w_h(r) = \frac{\alpha}{h^2} f(q) \tag{4}$$

où $q = |\underline{r} - \underline{r}'| / h$. Avec la condition de normalisation, nous obtenons

$$2\pi\alpha \int_0^{h_t} f(q)qdq = 1 \tag{5}$$

Cette condition permet de déterminer la constante α, connaissant la fonction f. Les trois noyaux disponibles dans SPARTACUS-2D sont les suivants (« *spline based kernels* » d'ordres respectifs 3, 4 et 5) :

---

[2] Nous verrons plus loin (**§§ 1.1.4.2** et **1.1.4.3**) que les bords solides sont également modélisés par des particules.

[3] Le mot Allemand *kernel* est généralement utilisé pour le désigner dans la littérature internationale. En dimension 2, cette fonction a la dimension de l'inverse d'une surface.

$$f(q)=\begin{cases}1-\dfrac{3}{2}q^2+\dfrac{3}{4}q^3 & si \quad 0\le q\le 1\\ \dfrac{1}{4}(2-q)^3 & si \quad 1\le q\le 2\\ 0 & si \quad 2\le q\end{cases} \tag{6}$$

$$f(q)=\begin{cases}\left(\dfrac{5}{2}-q\right)^4-5\left(\dfrac{3}{2}-q\right)^4+10\left(\dfrac{1}{2}-q\right)^4 & si \quad 0\le q\le 0{,}5\\ \left(\dfrac{5}{2}-q\right)^4-5\left(\dfrac{3}{2}-q\right)^4 & si \quad 0{,}5\le q\le 1{,}5\\ \left(\dfrac{5}{2}-q\right)^4 & si \quad 1{,}5\le q\le 2{,}5\\ 0 & si \quad 2{,}5\le q\end{cases} \tag{7}$$

$$f(q)=\begin{cases}(3-q)^5-6(2-q)^5+15(1-q)^5 & si \quad 0\le q\le 1\\ (3-q)^5-6(2-q)^5 & si \quad 1\le q\le 2\\ (3-q)^5 & si \quad 2\le q\le 3\\ 0 & si \quad 3\le q\end{cases} \tag{8}$$

La constante α apparaissant dans (4) vaut alors respectivement 10 / 7π pour (6), 96 / 1199π ($\approx$ 51 / 2000) pour (7) et 7 / 478π pour (8), afin de vérifier la condition de normalisation[4] (2). Ces noyaux sont continûment dérivables, et les rayons de leurs supports respectifs correspondent à des valeurs de h_t / h respectivement égales à 2, 2,5 et 3, rapport que nous noterons α_t. L'intégrale (3) est ensuite approchée par une somme de Riemann, l'erreur demeurant du même ordre de grandeur :

$$A(\underline{r})\approx\sum_b \frac{m_b}{\rho_b}A_b w_h\left(\left|\underline{r}-\underline{r}_b\right|\right) \tag{9}$$

la sommation étant étendue à toutes les particules b. Nous voyons que l'élément de volume dr' a été remplacé par le volume de chaque particule, c'est-à-dire le rapport m_b / ρ_b. Ainsi, la valeur du champ au point occupé par la particule a peut être mise sous la forme approchée :

$$A(\underline{r}_a)\approx\sum_b \frac{m_b}{\rho_b}A_b w_h(r_{ab}) \tag{10}$$

la quantité r_{ab} étant la norme du vecteur $\underline{r}_{ab}$ reliant les particules a et b, c'est-à-dire $\left|\underline{r}_a-\underline{r}_b\right|$. Notons dès à présent que le vecteur $\underline{r}_{ab}$ est dirigé de b vers a. Cette méthode d'interpolation montre que le champ A au point $\underline{r}$ dépend des valeurs que prend A aux points occupés par les différentes particules qui l'entourent, et que l'influence d'une particule sur ce point de l'espace diminue avec la distance[5]. Il

[4] Ces noyaux sont encore valables en dimension 3, mais avec des constantes α différentes et un exposant 3 au dénominateur de (4) ; w_h possède alors en effet la dimension de l'inverse d'un volume, et la condition de normalisation (5) portant sur f s'en trouve modifiée.

[5] Intuitivement, on peut comprendre le noyau comme une mesure de l'interaction entre deux particules. Formellement, cette

est cependant important de noter que la formule (10) donne une estimation $A(\underline{r}_a)$ en principe différente de A_a.

Du fait de la largeur finie du support du noyau, nous voyons que la contribution de chaque particule b à la valeur du champ A au point a s'annule pour les particules suffisamment éloignées (pour être exact, dès que $r_{ab} > h_t$, comme on le voit sur la **figure 2**). Ainsi, la particule a n'interagit qu'avec un nombre fini de particules b situées à l'intérieur d'un cercle de rayon h_t centrée sur elle.

C'est à ces seules particules que la somme (10) doit donc être étendue. Pour les noyaux donnés par les formules (6), (7) et (8), en considérant un rapport $\alpha_h = h/\delta r = 1{,}5$ (valeur typique et recommandée), le nombre N_v de voisines d'une particule donnée est alors à peu près $\pi(\alpha_t \alpha_h)^2$ (en dimension 2), c'est-à-dire de l'ordre de 28 à 64 suivant l'ordre du noyau, avec de légères variations dépendant de la disposition locale des particules.

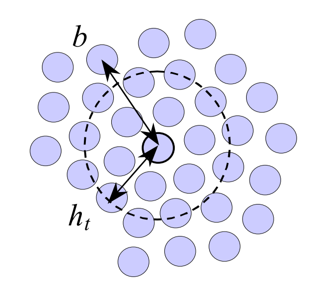


Figure 2 – Domaine d'influence de la particule *a* située au centre (ici, *b* n'agit pas sur *a*).

1.1.1.2 Différentiation

Nous voyons à présent que l'équation (10) peut être dérivée formellement par rapport aux coordonnées d'espace de la particule a. Ainsi, le gradient du champ scalaire A pris au point occupé par la particule a peut s'écrire de manière approchée

$$\nabla A(\underline{r}_a) \approx \sum_b \frac{m_b}{\rho_b} A_b \nabla_a w_h(r_{ab}) \quad (11)$$

la quantité $\nabla_a w_h(r_{ab})$ désignant le gradient du noyau par rapport aux coordonnées de a, directement relié à la fonction f définie dans (4) par

$$\nabla_a w_h(r_{ab}) = \frac{\alpha}{h^3} f'(q_{ab}) \underline{e}_{ab} \quad (12)$$

q_{ab} représentant la valeur de q associée à la paire de particules a et b (c'est-à-dire r_{ab} / h) et $\underline{e}_{ab} \equiv \underline{r}_{ab} / r_{ab}$ étant le vecteur unitaire dirigé de b vers a. L'expression (11) constitue une formulation possible du gradient d'un scalaire dans le formalisme SPH. On voit qu'il n'est pas besoin de grille pour évaluer le gradient d'un champ scalaire, puisqu'il s'exprime en fonction du gradient du noyau, dont la formulation analytique est connue ; les seules valeurs du champ aux points occupés par les particules voisines suffisent. Compte tenu des noyaux donnés par (6), (7) et (8), nous avons respectivement :

fonction joue d'autre part un rôle analogue aux fonctions de base dans le formalisme de la méthode des éléments finis.

$$f'(q)=\begin{cases}\frac{9}{4}q^2-3q & si \quad 0\le q\le 1\\ -\frac{3}{4}(2-q)^2 & si \quad 1\le q\le 2\\ 0 & si \quad 2\le q\end{cases} \tag{13}$$

$$f'(q)=-4\begin{cases}\left(\frac{5}{2}-q\right)^3-5\left(\frac{3}{2}-q\right)^3+10\left(\frac{1}{2}-q\right)^3 & si \quad 0\le q\le 0{,}5\\ \left(\frac{5}{2}-q\right)^3-5\left(\frac{3}{2}-q\right)^3 & si \quad 0{,}5\le q\le 1{,}5\\ \left(\frac{5}{2}-q\right)^3 & si \quad 1{,}5\le q\le 2{,}5\\ 0 & si \quad 2{,}5\le q\end{cases} \tag{14}$$

$$f'(q)=-5\begin{cases}(3-q)^4-6(2-q)^4+15(1-q)^4 & si \quad 0\le q\le 1\\ (3-q)^4-6(2-q)^4 & si \quad 1\le q\le 2\\ (3-q)^4 & si \quad 2\le q\le 3\\ 0 & si \quad 3\le q\end{cases} \tag{15}$$

Les noyaux (6), (7) et (8), ainsi que leurs gradients respectifs (13), (14) et (15) sont représentés en **figure 3**. On y peut observer que la dérivée du noyau est toujours négative ; ainsi, $\nabla_a w_h(r_{ab})$ est un vecteur dirigé de *a* vers *b*. Nous pouvons aussi affirmer, pour faire écho au commentaire de la fin du **§ 1.1.1.1**, que l'interaction entre particules s'affaiblit avec la distance, résultat conforme à l'intuition qui justifie le choix des noyaux retenus ici.

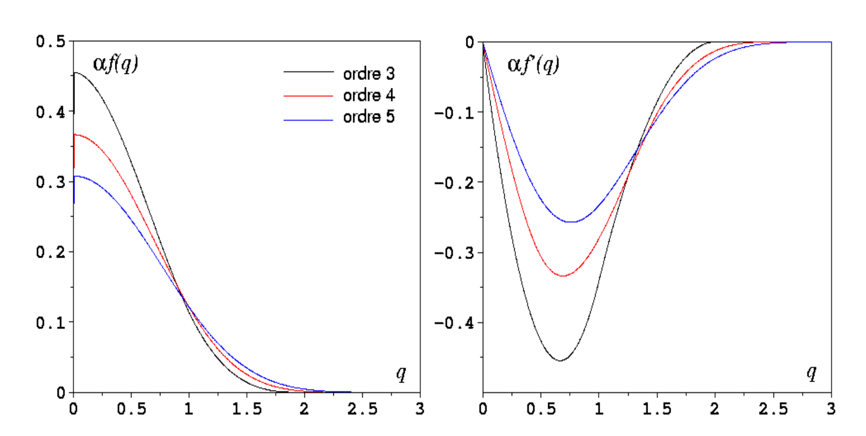


Figure 3 – Noyaux α*f*(*q*) utilisés dans SPARTACUS-2D,
Ainsi que leurs gradients α*f'*(*q*).

Ce qui vient d'être dit pour l'opérateur gradient demeure vrai pour n'importe quel opérateur différentiel du premier ordre. Ainsi, un calcul simple permet de montrer que la divergence d'un champ de vecteurs $\underline{A}$ peut s'écrire, dans le même formalisme :

$$\nabla \cdot \underline{A}(\underline{r}_a) \approx \sum_b \frac{m_b}{\rho_b} \underline{A}_b \cdot \nabla_a w_h(r_{ab}) \tag{16}$$

Cependant, ainsi que nous le verrons plus loin, les formulations (11) et (16) ne sont pas les plus appropriées. Tout comme les méthodes de type « volumes finis » et « éléments finis », la méthode SPH autorise plusieurs formes pour chaque opérateur différentiel. Ainsi, en écrivant l'équation exacte

$$\nabla A = \rho \nabla \frac{A}{\rho} + \frac{A}{\rho} \nabla \rho \tag{17}$$

(où ρ est la densité) et en appliquant l'approximation (11) à chacun des gradients apparaissant dans le membre de droite de (17), nous aboutissons à une autre forme d'opérateur gradient :

$$\nabla A(\underline{r}_a) \approx \rho_a \sum_b m_b \left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2} \right) \nabla_a w_h(r_{ab}) \tag{18}$$

Au contraire de l'approximation (11), cette dernière présente un caractère antisymétrique par rapport aux indices *a* et *b*, ce qui est un avantage fondamental, ainsi que nous le verrons plus loin (Cf. **§ 1.1.2**). La forme (18) n'est pas la seule présentant cette propriété ; des manipulations similaires à (17) permettent notamment d'écrire une autre forme SPH du gradient :

$$\nabla A(\underline{r}_a) \approx \rho_a \sum_b m_b \frac{A_a + A_b}{\rho_a \rho_b} \nabla_a w_h(r_{ab}) \tag{19}$$

Il n'est pas évident de se décider en faveur de (19) plutôt que (18), bien que des considérations variationnelles permettent d'éclaircir la question, sur lesquelles nous reviendrons brièvement au **§ 1.1.2.3**. De la même manière, le gradient (tensoriel) d'une quantité vectorielle peut s'écrire de plusieurs façons, mais nous retiendrons dans SPARTACUS-2D la forme suivante :

$$\nabla \underline{A}(\underline{r}_a) \approx -\frac{1}{\rho_a} \sum_b m_b \underline{A}_{ab} \otimes \nabla_a w_h(r_{ab}) \tag{20}$$

où ⊗ désigne le produit tensoriel de deux vecteurs et $\underline{A}_{ab} = \underline{A}_a - \underline{A}_b$. La forme (20) possède l'avantage de donner un gradient identiquement nul pour un champ de vecteurs constant.

Toujours en se fondant sur la même méthode, après une manipulation simple, il est possible de déterminer une forme d'opérateur divergence préférable à (16) :

$$\nabla \cdot \underline{A}(\underline{r}_a) \approx -\frac{1}{\rho_a} \sum_b m_b \underline{A}_{ab} \cdot \nabla_a w_h(r_{ab}) \tag{21}$$

présentant à nouveau une symétrie. Une relation similaire à (17) donnerait enfin la forme suivante :

$$\nabla \cdot \underline{A}(\underline{r}_a) \approx \rho_a \sum_b m_b \left(\frac{\underline{A}_a}{\rho_a^2} + \frac{\underline{A}_b}{\rho_b^2} \right) \cdot \nabla_a w_h(r_{ab}) \tag{22}$$

Parmi les formes proposées plus haut pour les opérateurs gradient et divergence dans le langage SPH, nous aurons à faire des choix ; ces choix, ainsi que leurs raisons et les conséquences qui en résultent, seront exposés au **§ 1.1.2**.

Nous aurons enfin besoin d'une forme SPH de l'opérateur laplacien. Il semblerait judicieux, pour ce

faire, d'appliquer formellement cet opérateur à l'interpolation (10) ; la forme obtenue ferait cependant intervenir le laplacien du noyau, qui a une fâcheuse tendance à se comporter de manière trop sensible au désordre particulaire, générant des instabilités numériques (Cf. Issa, 2005). Pour cette raison, il est préférable d'adopter une forme approchée fondée sur les seules dérivées premières du noyau. Parmi les formes proposées dans la littérature, nous en adopterons deux (sans les démontrer), construites à partir des formules respectivement données par Monaghan (1992) et Morris *et al.* (1997) :

$$\nabla \cdot \left(B\nabla \underline{A}\right)\!\left(\underline{r}_a\right) \approx 8\sum_b m_b \frac{B_a + B_b}{\rho_a + \rho_b} \frac{\underline{A}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2} \nabla_a w_h\left(r_{ab}\right) \tag{23}$$

$$\nabla \cdot \left(B\nabla \underline{A}\right)\!\left(\underline{r}_a\right) \approx \sum_b m_b \frac{\rho_a B_a + \rho_b B_b}{\rho_a \rho_b} \frac{\underline{A}_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right) \tag{24}$$

Nous avons ici, plutôt que des laplaciens, des opérateurs de *diffusion* fondés sur un coefficient de diffusion B éventuellement variable, ce qui sera d'une grande utilité pour la modélisation de la turbulence (Cf. § **1.1.3.3**). Les formes (23) et (24), bien que donnant des approximations presque identiques, n'ont pas tout à fait les mêmes propriétés physiques, comme on le verra au **§ 1.1.2.3**.

Pour la diffusion d'un scalaire, enfin, nous privilégions ici une forme analogue à (24) :

$$\nabla \cdot \left(B\nabla A\right)\!\left(\underline{r}_a\right) \approx \sum_b m_b \left(\frac{\rho_a B_a + \rho_b B_b}{\rho_a \rho_b}\right) \frac{A_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right) \tag{25}$$

avec $A_{ab} = A_a - A_b$. Il faut noter que les équations (23) à (25) présentent une certaine limitation, car la distance r_{ab} y apparaît au dénominateur. Afin d'éviter les divisions par zéro (dans le cas accidentel où deux particules coïncideraient), l'usage (adopté dans SPARTACUS-2D) est de remplacer r_{ab}^2 par $r_{ab}^2 + \eta^2$, où $\eta = 0{,}1h$ est un petit paramètre de *clipping*. Pour alléger les notations, nous omettrons cependant cette correction dans la suite de ce document.

Ainsi, nous constatons que tous les opérateurs différentiels nécessaires à l'établissement des équations de la mécanique des fluides peuvent être discrétisés dans le contexte de la méthode SPH, et ce à l'aide du seul gradient du noyau. Les outils numériques présentés dans ce paragraphe vont servir à présent de base à l'écriture des équations de la mécanique des fluides dans le formalisme SPH.

1.1.2 Ecriture des équations pour la mécanique des fluides

1.1.2.1 Equation de continuité

A partir de la méthode de différentiation explicitée au paragraphe précédent, nous pouvons établir une formulation de l'équation de continuité d'un fluide compressible. Pour cela, écrivons cette équation dans un formalisme lagrangien, en écrivant la dérivée particulaire de la densité sous la forme

$$\frac{d\rho}{dt} = -\rho\nabla \cdot \underline{u} \tag{26}$$

En utilisant l'approximation (22) pour le membre de droite, nous obtiendrions

$$\frac{d\rho_a}{dt} = -\rho_a^2 \sum_b m_b \left(\frac{\underline{u}_a}{\rho_a^2} + \frac{\underline{u}_b}{\rho_b^2}\right) \cdot \nabla_a w_h\left(r_{ab}\right) \tag{27}$$

L'une des propriétés essentielles de l'approche SPH est de conserver la masse par nature, puisque les masses des particules sont choisies constantes. En revanche, le fait de travailler dans un formalisme compressible signifie que le volume n'est *a priori* pas conservé. Examinons en effet la variation de volume d'une particule a :

$$\frac{d\tau_a}{dt} = \frac{dm_a / \rho_a}{dt} = -\frac{m_a}{\rho_a^2}\frac{d\rho_a}{dt} = \tau_a \left(\nabla \cdot \underline{u}\right)_a \qquad (28)$$

(on a utilisé le fait que la masse est constante et l'équation (26)). La dernière égalité, bien connue, signifie que la divergence du champ de vitesse représente localement le *taux de variation volumique*, c'est-à-dire la vitesse de variation d'un volume élémentaire, ramenée à la mesure de ce volume. Or, l'équation (27) nous montre que la contribution d'une particule b à cette variation s'écrit :

$$q_{V,ba} = m_a m_b \left(\frac{\underline{u}_a}{\rho_a^2} + \frac{\underline{u}_b}{\rho_b^2}\right) \cdot \nabla_a w_h\left(r_{ab}\right) \qquad (29)$$

Nous avons appelé cette grandeur $q_{V,ba}$ car elle représente un *flux* de volume dirigé de b vers a, c'est-à-dire le volume cédé à a par b par unité de temps. Or, en raison de la symétrie du noyau, son gradient est au contraire antisymétrique par rapport aux indices a et b :

$$\nabla_b w_h\left(r_{ba}\right) = -\nabla_a w_h\left(r_{ab}\right) \qquad (30)$$

Nous constatons alors que le flux de volume (29) est *antisymétrique*. Cela signifie que le volume cédé par b à a est intégralement récupéré par a ($q_{V,ab} = -q_{V,ba}$), autrement dit que le volume total est conservé. Or, ce résultat intéressant n'est ici pas physique puisque nous travaillons dans un contexte de fluide compressible. Nous voyons que, bien que la divergence du champ de vitesse dans (27) ne soit pas nécessairement nulle, c'est-à-dire bien que le volume soit localement non conservé, il l'est en revanche globalement, ce qui n'est pas conforme à la réalité. Ces considérations seront d'une grande importance dans la suite, car toutes les autres équations retenues dans SPARTACUS-2D présentent des propriétés de symétrie analogues à (27), aboutissant généralement à des lois de conservation *exactes et physiques*. Pourtant, dans le cas de l'équation de continuité, ce qui précède montre que nous devons adopter une autre forme. Si nous reprenons (26) mais en approchant la divergence du champ de vitesse grâce à (21) plutôt que (22), nous obtenons

$$\frac{d\rho_a}{dt} = \sum_b m_b \underline{u}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right) \qquad (31)$$

où l'on a posé $\underline{u}_{ab} = \underline{u}_a - \underline{u}_b$. Le raisonnement précédent ne permet alors pas d'aboutir à la même conclusion, en raison de la *symétrie* du terme de droite. Qui plus est, nous constatons que la divergence de la vitesse s'annule à présent si les particules possèdent toutes la même vitesse, ce qui est cohérent. Nous sommes donc en présence d'une forme de l'équation de continuité meilleure que (27), comme va le confirmer le raisonnement qui suit.

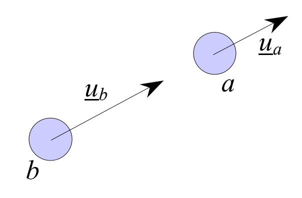


Figure 4 – Echange de masse entre deux particules.

Considérons deux particules a et b, b se dirigeant vers a avec une vitesse supérieure (voir **figure 4**), si bien que $\underline{u}_{ab}$ est un vecteur aligné avec $\underline{r}_{ab}$, dirigé vers b. Nous avons vu au **§ 1.1.1.2** que $\nabla_a w_h\left(r_{ab}\right)$ est

également dirigé de *a* vers *b* ; ainsi la contribution de *b* à la masse volumique de *a* est positive, c'est-à-dire que *a* reçoit du volume de la part de *b*, ce qui est encore une fois conforme à l'intuition physique. C'est l'inverse qui se passe dans le cas où la vitesse de *b* est inférieure à celle de *a*, ce dont rend compte la forme (31), tandis que la forme (27) donnerait dans ce cas un flux de volume dirigé de *b* vers *a*, indépendamment du signe de $\underline{u}_{ab}$.

Nous retiendrons donc la forme (31) pour l'équation de continuité. L'approche proposée possède l'avantage de la simplicité et présente en outre des propriétés de stabilité vis-à-vis des surfaces libres (Cf. Monaghan, 1994). L'équation (31) est la première équation du modèle SPH « standard ». Elle est utilisée dans SPARTACUS-2D pour évaluer la densité de chaque particule à chaque pas de temps, à l'aide d'un schéma numérique explicité plus loin (Cf. **§ 1.2.1.1**). La connaissance du champ de vitesse à un instant donné permet donc de réactualiser les densités au même instant ou à l'instant suivant (des détails sur le schéma numérique seront donnés au **§ 1.2.1.2**).

A la lumière de ce qui précède, nous voyons que le choix d'une forme SPH pour tel ou tel opérateur différentiel n'est pas anodin ; comme nous le verrons encore dans la suite de cette partie, les propriétés des différentes formes présentées au **§ 1.1.1.2** sont extrêmement variées.

1.1.2.2 Equation de quantité de mouvement

Nous allons à présent chercher à écrire une forme SPH de l'équation de quantité de mouvement de Navier-Stokes incompressible[6]. Pour cela, nous commençons par en écrire une forme lagrangienne, en joignant le terme de convection des vitesses au terme de dérivée temporelle, pour former l'accélération (dérivée lagrangienne) de chaque parcelle de fluide :

$$\frac{d\underline{u}}{dt} = -\frac{1}{\rho}\nabla p + \nu\Delta\underline{u} + \underline{F}^e \qquad (32)$$

Dans cette équation, la pression est désignée par *p*, tandis que la viscosité cinématique du fluide est notée ν. Le champ de vecteurs $\underline{F}^e$ fait référence aux forces extérieures par unité de masse, en particulier la gravité ou une force motrice horizontale représentant l'effet de l'inclinaison d'un canal (Cf. **§ 2.4.2.3**). Nous verrons au **§ 1.1.4.2** que d'autres forces extérieures peuvent être appliquées à telle ou telle particule pour modéliser l'effet des parois solides.

Par un procédé semblable à celui du paragraphe précédent, et en faisant appel aux approximations (18) et (24), la forme SPH de cette équation prend alors l'allure suivante :

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} - \nu \frac{\rho_a + \rho_b}{\rho_a \rho_b} \frac{\underline{u}_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \right) \nabla_a w_h(r_{ab}) + \underline{F}_a^e \qquad (33)$$

le terme $\nu\Delta\underline{u}$ se présentant sous la forme plus générale $\nabla\cdot(\nu\nabla\underline{u})$, bien que la viscosité soit ici constante (cela ne sera en revanche plus le cas en régime turbulent, dès le **§ 1.1.3**). L'équation (33) est la deuxième du modèle SPH, et est utilisée dans SPARTACUS-2D pour calculer les vitesses des

[6] Comme nous venons de le voir (Cf **§ 1.1.2.1**), la méthode SPH est en réalité applicable aux seuls fluides *compressibles*. Toutefois, dans le cas de fluides quasi-incompressibles, nous pouvons faire l'hypothèse raisonnable que les termes additionnels de l'équation de quantité de mouvement (dépendant de la divergence du champ de vitesse) sont négligeables devant les termes considérés ici.

particules à chaque pas de temps. Si l'on choisissait en lieu et place de (18) et (24) les formules (19) et (23) pour modéliser le gradient de pression et le terme visqueux, on aboutirait alors à

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{16\nu}{\rho_a + \rho_b} \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2} \right) \nabla_a w_h(r_{ab}) + \underline{F}_a^e \tag{34}$$

qui en est une autre forme possible. Nous pourrions aussi bien croiser les formes retenues (c'est-à-dire combiner la forme du gradient de pression de (33) avec le terme visqueux de (34), et inversement). D'une manière générale, l'équation de quantité de mouvement pour chaque particule s'écrit

$$\frac{d\underline{u}_a}{dt} = \sum_b m_b \left(\underline{q}_{p,ba} + \underline{q}_{\nu,ba} \right) + \underline{F}_a^e \tag{35}$$

les quantités $\underline{q}_{p,ba}$ et $\underline{q}_{\nu,ba}$ représentant des *flux d'impulsion* de b vers a, associés respectivement aux forces de pression et de viscosité (par unité de masse) ; ils peuvent prendre indifféremment les formes proposées en (33) et (34)[7]. Dans tous les cas, la connaissance des champs de vitesse et de pression à un instant donné permet le calcul des vitesses à l'instant suivant ; quelle que soit sa forme, l'équation de quantité de mouvement nécessite donc (comme l'équation de continuité (31)) un schéma numérique (Cf. **§ 1.2.1.2**). Quant aux positions des particules, elles sont calculées en intégrant les vitesses à l'aide du même schéma, à partir de l'équation

$$\frac{d\underline{r}_a}{dt} = \underline{u}_a \tag{36}$$

Bien entendu, l'intégration des équations (31), (33) (ou (34)) et (36) requiert des conditions aux limites appropriées, qui seront présentées au **§ 1.1.4**.

Nous remarquons que le terme situé sous le signe somme du membre de droite de (33) et (34) (c'est-à-dire le flux total d'impulsion) est *antisymétrique* vis-à-vis des indices a et b, car les expressions (18), (19), (23) et (24) le sont, grâce à la propriété (30). La quantité

$$\underline{q}_{u,ba} = m_a m_b \left(\underline{q}_{p,ba} + \underline{q}_{\nu,ba} \right) = \underline{F}_{b\to a} \tag{37}$$

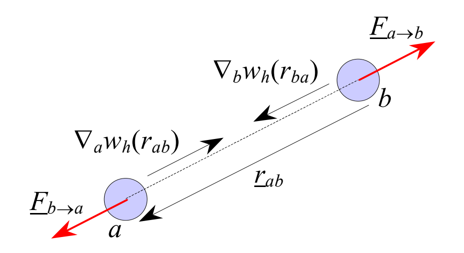


Figure 5 – Principe d'action-réaction.

représente le flux d'impulsion de b vers a, c'est-à-dire la *force* communiquée par b à a. Cette symétrie signifie donc que les forces exercées entre particules vérifient le principe d'action-réaction (Cf. **figure 5**), c'est-à-dire que le formalisme SPH assure la conservation exacte de l'impulsion totale du fluide (nous verrons au **§ 1.1.2.3** l'origine profonde de cette intéressante propriété). D'autre part, le vecteur $\underline{q}_{p,ba}$ est aligné avec $\underline{r}_{ab}$ et dirigé de b vers a, quel que soit le modèle retenu, ce qui est conforme aux lois physiques.

---

[7] … en principe. Des commentaires seront néanmoins faits au **§ 1.1.2.4**, concernant les propriétés respectives des deux modèles de viscosité proposés.

Notons enfin que l'équation de quantité de mouvement est invariante par transformation galiléenne. Le terme visqueux se comporte cependant de manière différente suivant la forme adoptée : le modèle de $\underline{q}_{\nu,ba}$ donné dans (34) présente un produit scalaire $\underline{u}_{ab} \cdot \underline{r}_{ab}$ qui rend le terme visqueux nul dans le cas d'un mouvement de rotation solide, ce qui est conforme aux principes sur lesquels le terme mathématique $\nu\Delta\underline{u}$ est établi pour modéliser les effets visqueux dans l'équation de Navier-Stokes (32). Cela n'est pas le cas du modèle donné dans (33), qui aboutit à une force visqueuse alignée avec $\underline{u}_{ab}$. Cependant, cette dernière forme semble plus en accord avec la notion de cisaillement dans un milieu continu élastique linéaire. Nous verrons pourtant au **§ 1.1.2.4** que le modèle (34) possède encore un autre avantage.

Enfin, il est important de noter que la force qu'une particule exerce sur elle-même est *nulle* (propriété qui répond à une notion physique), car le gradient du noyau s'annule pour $r_{ab} = 0$, comme on le voit en examinant les équations (13) à (15). Cela permet de s'affranchir de la particule a dans le calcul des sommes intervenant dans (31), (33) et (34).

1.1.2.3 Remarques sur les propriétés de conservation des équations

La propriété de conservation (concernant la quantité de mouvement) que nous avons établie au paragraphe précédent n'est pas fortuite. Nous allons brièvement revenir sur les lois fondamentales de la mécanique, afin d'expliquer précisément la raison et la signification de cette conservation. Pour cela, nous renvoyons le lecteur à l'ouvrage de Landau et Lifchitz (1981) ou à Violeau (2000), où l'on trouvera un exposé détaillé des fondements mathématiques de la mécanique du point. Pour les besoins de la discussion qui suit, nous retiendrons tout d'abord qu'un système dynamique peut être caractérisé par une « fonction d'état » L nommé *lagrangien*, se ramenant pour un système de points matériels en interaction à la différence entre énergie cinétique et énergie potentielle (ne dépendant que des positions des particules) :

$$L = \sum_a \frac{1}{2} m_a u_a^2 - U\left(\underline{r}_a, t\right) \qquad (37)$$

D'autre part, un principe variationnel nommé *principe de moindre action* permet d'affirmer *in fine* que le lagrangien satisfait aux équations de Lagrange

$$\frac{d}{dt}\frac{\partial L}{\partial \underline{u}_a} = \frac{\partial L}{\partial \underline{r}_a} \qquad (38)$$

valables pour chaque particule, et constituant les équations du mouvement du système. A partir de la forme (38), nous pouvons aisément en déduire l'équation régissant le mouvement des constituants du système :

$$m_a \frac{d\underline{u}_a}{dt} = -\frac{\partial U}{\partial \underline{r}_a} = \underline{F}_a \qquad (39)$$

Dans cette dernière équation (qui n'est autre que la loi du mouvement de Newton), nous voyons apparaître la *force* exercée sur la particule a, définie comme le gradient de l'énergie potentielle. Des considérations simples (mais essentielles) permettent alors de montrer les propriétés suivantes :

- L'invariance des lois de la mécanique dans le temps aboutit à la conservation de l'énergie totale d'un système fermé, définie comme la somme des énergies cinétique et potentielle.

- De l'homogénéité de l'espace vis-à-vis des lois de la mécanique résulte la conservation de la quantité de mouvement (ou impulsion) totale d'un système fermé, définie comme la somme des impulsions individuelles $m_a\underline{u}_a$.
- Enfin, l'isotropie de l'espace est associée à la conservation du moment cinétique total d'un système fermé, défini comme la somme des quantités $\underline{r}_a \times m_a\underline{u}_a$.

Ces généralités étant admises, nous allons à présent examiner les conséquences pour un système fluide constitué de particules représentant de petits volumes macroscopiques. Le rôle de l'énergie potentielle U est alors joué par l'énergie interne E_{int}, à laquelle vient s'ajouter l'énergie potentielle de gravité $m_a g z_a$, z_a étant l'altitude de la particule a. Ainsi, l'équation (39) se ramène à

$$m_a \frac{d\underline{u}_a}{dt} = -\sum_b m_b \frac{\partial e_b}{\partial \underline{r}_a} + m_a \underline{g} \tag{40}$$

avec $\underline{g} = -g\underline{e}_z$, $\underline{e}_z$ étant le vecteur unitaire ascendant, et e_b l'énergie interne de la particule b par unité de masse (si bien que l'énergie interne totale du système est écrite comme la somme des $m_b e_b$). A présent, nous allons estimer le gradient de cette énergie interne en faisant appel au formalisme SPH. Pour cela, commençons par écrire

$$\frac{\partial e_b}{\partial \underline{r}_a} = \frac{\partial e_b}{\partial \rho_b} \frac{\partial \rho_b}{\partial \underline{r}_a} \tag{41}$$

Puis, en utilisant l'équation (10), nous avons

$$\rho_b \approx \sum_c m_c w_h(r_{bc}) \tag{42}$$

si bien que le gradient de la densité s'écrit

$$\frac{\partial \rho_b}{\partial \underline{r}_a} = m_a \nabla_a w_h(r_{ba}) + \delta_{ab} \sum_c m_c \nabla_a w_h(r_{ac}) \tag{43}$$

Dans cette formule, le premier terme est pris pour $c = a$, tandis que le second n'intervient que si b se trouve être égal à a (d'où la présence du symbole de Kronecker δ_{ab}). En revenant à l'équation (41), la force apparaissant dans (40) donne alors

$$\begin{aligned}
\sum_b m_b \frac{\partial e_b}{\partial \underline{r}_a} &= \sum_b m_a m_b \left(\frac{\partial e}{\partial \rho}\right)_b \nabla_a w_h(r_{ba}) + \sum_{b,c} m_c m_b \left(\frac{\partial e}{\partial \rho}\right)_b \delta_{ab} \nabla_a w_h(r_{ac}) \\
&= \sum_b m_a m_b \left(\frac{\partial e}{\partial \rho}\right)_b \nabla_a w_h(r_{ab}) + \sum_c m_c m_a \left(\frac{\partial e}{\partial \rho}\right)_a \nabla_a w_h(r_{ac}) \\
&= \sum_b m_a m_b \left[\left(\frac{\partial e}{\partial \rho}\right)_b + \left(\frac{\partial e}{\partial \rho}\right)_a\right] \nabla_a w_h(r_{ab})
\end{aligned} \tag{44}$$

(nous avons utilisé dans ce bref calcul la symétrie du noyau et la définition de δ_{ab}). Nous constatons que cette force est bien antisymétrique par rapport aux indices a et b, ce qui reflète la conservation de l'impulsion totale. Pour conclure, nous devons faire appel à la thermodynamique. La relation $dE_{int} = pdV$ (où V est le volume) s'écrit encore $de = -pd\rho / \rho^2$, ce qui signifie que $-p / \rho^2$ représente la dérivée de l'énergie par rapport à la densité, apparaissant dans (44). Finalement, en divisant par m_a, l'équation du mouvement (40) s'écrit pour chaque particule :

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \nabla_a w_h(r_{ab}) + \underline{g} \qquad (45)$$

Cette équation n'est autre que l'équation du mouvement du modèle SPH (33), où l'on n'a considéré que les forces de pression, la gravité jouant le rôle des forces extérieures. L'antisymétrie de cette équation apparaît alors comme une conséquence du fait qu'elle dérive directement d'un principe de moindre action, satisfaisant dès lors les lois de conservation annoncées plus haut. On vérifiera en effet que le moment cinétique est conservé par (45) pour un système fermé (donc non soumis à la gravité) :

$$\begin{aligned} \frac{d}{dt}\sum_a \underline{r}_a \times m_a \underline{u}_a &= \sum_a \frac{d\underline{r}_a}{dt} \times m_a \underline{u}_a + \sum_a \underline{r}_a \times m_a \frac{d\underline{u}_a}{dt} \\ &= \sum_a m_a \underline{u}_a \times \underline{u}_a - \sum_{a,b} m_a m_b \underline{r}_a \times \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \nabla_a w_h(r_{ab}) \end{aligned} \qquad (46)$$

Dans le membre de droite de cette égalité, le premier terme est nul car le produit vectoriel d'un vecteur avec lui-même est nul, tandis que le second peut être séparé en deux contributions, selon que *a* est plus petit que *b* ou inversement :

$$\begin{aligned} \frac{d}{dt}\sum_a \underline{r}_a \times m_a \underline{u}_a &= -\sum_{a<b} m_a m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \underline{r}_a \times \nabla_a w_h(r_{ab}) - \sum_{a>b} m_a m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \underline{r}_a \times \nabla_a w_h(r_{ab}) \\ &= -\sum_{a<b} m_a m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \underline{r}_a \times \nabla_a w_h(r_{ab}) - \sum_{a<b} m_a m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \underline{r}_b \times \nabla_a w_h(r_{ab}) \\ &= -\sum_{a<b} m_a m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \underline{r}_{ab} \times \nabla_a w_h(r_{ab}) \\ &= \underline{0} \end{aligned} \qquad (47)$$

(on a effectué un changement d'indices muets *a* et *b*, puis utilisé la symétrie du noyau et le fait que le gradient de ce dernier est aligné avec $\underline{r}_{ab}$). Remarquons que les résultats donnés par (46) et (47) ne font intervenir que la symétrie du terme de pression, et demeureraient donc vrais pour une forme du gradient de pression similaire à (34). De fait, un traitement approprié du terme (42) dans le raisonnement fait plus haut permettrait de déduire d'un principe de moindre action toute forme symétrique du terme de pression (comme celle donnée par l'équation (34)).

Quant à l'énergie, sa conservation provient de sa définition et de l'équation du mouvement, sans même faire intervenir l'approximation SPH :

$$\begin{aligned} \frac{d}{dt}\left[\sum_a \frac{1}{2} m_a u_a^2 + U(\underline{r}_a, t) \right] &= \sum_a \underline{u}_a \cdot m_a \frac{d\underline{u}_a}{dt} + \frac{d}{dt} U(\underline{r}_a, t) \\ &= -\sum_a \underline{u}_a \cdot \frac{\partial U}{\partial \underline{r}_a} + \frac{\partial U}{\partial t} + \sum_a \frac{d\underline{r}_a}{dt} \cdot \frac{\partial U}{\partial \underline{r}_a} \end{aligned} \qquad (48)$$

Les deux termes extrêmes sont opposés, tandis que le terme central s'annule pour un système fermé (c'est-à-dire ici ne subissant aucune influence extérieure variable dans le temps).

1.1.2.4 Remarques sur les forces visqueuses

Examinons à présent le comportement du terme visqueux vis-à-vis des lois de conservation introduites

précédemment. Il est impossible d'obtenir les forces visqueuses d'un fluide par la méthode exposée au paragraphe précédent, car nous savons qu'elles violent la conservation de l'énergie, en tant que modélisant un processus irréversible. Il nous faut donc modifier l'équation de Lagrange (38) par la méthode suivante. Remarquons tout d'abord que les forces de frottement dépendent des vitesses, ou plus exactement des différences de vitesses. On sait également que les frottements au sein d'un fluide newtonien en dépendent linéairement[8], si bien qu'il est permis d'écrire sous forme très générale, en se référant aux notations de la formule (37) :

$$\underline{F}_{\nu,b\to a} = m_a m_b \underline{\tilde{q}}_{\nu,ba} = -\underline{\underline{\alpha}}_{ab}\,\underline{u}_{ab} \qquad (49)$$

ce qui est la forme la plus générale d'une dépendance linéaire, $\underline{\underline{\alpha}}_{ab}$ étant un tenseur du second ordre. La quantité (49) désigne la force de frottement exercée par *b* sur *a*, et le principe d'action-réaction nous enseigne qu'elle doit être antisymétrique, ce qui signifie que $\underline{\underline{\alpha}}_{ab}$ est symétrique vis-à-vis de *a* et *b*. La force de frottement totale reçue par *a* est donc

$$\underline{F}_{\nu,a} = -\sum_b \underline{\underline{\alpha}}_{ab}\,\underline{u}_{ab} \qquad (50)$$

Nous voyons que ce type de force ne dérive d'aucun potentiel, mais la symétrie des $\underline{\underline{\alpha}}_{ab}$ permet de l'écrire sous forme de la dérivée *par rapport aux vitesses* d'une fonction que nous appellerons *F* :

$$\underline{F}_{\nu,a} = -\frac{\partial F}{\partial \underline{u}_a} = -\frac{\partial}{\partial \underline{u}_a}\left(\frac{1}{2}\sum_{a,b} \underline{u}_{ab}\,\underline{\underline{\alpha}}_{ab}\,\underline{u}_{ab}\right) \qquad (51)$$

Prendre en compte cette force dans les équations de Lagrange donnerait finalement :

$$m_a \frac{d\underline{u}_a}{dt} = -\frac{\partial U}{\partial \underline{r}_a} - \frac{\partial F}{\partial \underline{u}_a} \qquad (52)$$

Nous allons tout d'abord montrer que cette force est effectivement dissipative, en reprenant le calcul (48) muni de cette nouvelle équation du mouvement :

$$\begin{aligned}\frac{d}{dt}\left[\sum_a \frac{1}{2} m_a u_a^2 + U(\underline{r}_a,t)\right] &= \sum_a \underline{u}_a \cdot m_a \frac{d\underline{u}_a}{dt} + \frac{d}{dt}U(\underline{r}_a,t) \\ &= -\sum_a \underline{u}_a \cdot \frac{\partial U}{\partial \underline{r}_a} - \sum_a \underline{u}_a \cdot \frac{\partial F}{\partial \underline{u}_a} + \frac{\partial U}{\partial t} + \sum_a \frac{d\underline{r}_a}{dt}\cdot\frac{\partial U}{\partial \underline{r}_a} \\ &= -\sum_a \underline{u}_a \cdot \sum_b \underline{\underline{\alpha}}_{ab}\,\underline{u}_{ab} = -\frac{1}{2}\sum_{a,b} \underline{u}_{ab}\,\underline{\underline{\alpha}}_{ab}\,\underline{u}_{ab} = -F\end{aligned} \qquad (53)$$

(pour achever ce calcul, nous avons séparé le terme de la troisième ligne en deux contributions identiques, puis opéré un changement d'indice muet et tiré avantage des propriétés de symétrie des différentes grandeurs). La fonction *F* joue donc le rôle de *taux de dissipation d'énergie* du système. Des considérations thermodynamiques permettent alors d'en déduire $F > 0$, c'est-à-dire que le tenseur

---

[8] Ceci vient du fait que les forces visqueuses sont très faibles en comparaison des forces de pression, si bien que les termes d'ordres supérieurs à 2 peuvent être négligés en développant les contraintes de Cauchy en série de Taylor par rapport aux vitesses. Dans le cas d'un modèle de fermeture turbulente fondé sur une approche de type « viscosité turbulente » (voir § **1.1.3.1**), cette propriété est mise en défaut pour les contraintes de Reynolds (voir la note du bas de la page 34).

$\underline{\underline{\alpha}}_{ab}$ est défini positif (vis-à-vis des indices a et b).

Il nous reste à considérer la conservation du moment cinétique. Des considérations analogues à celles du paragraphe précédent permettent de montrer que cette loi est satisfaite par les forces visqueuses (51) à condition que $\underline{\underline{\alpha}}_{ab}$ se mette sous la forme $\underline{r}_{ab} \otimes \underline{\beta}_{ab}$ avec $\underline{\beta}_{ba} = -\underline{\beta}_{ab}$, ce qui donne :

$$\underline{F}_{\nu,a} = -\sum_b \left(\underline{\beta}_{ab} \cdot \underline{u}_{ab}\right)\underline{r}_{ab} \tag{54}$$

C'est à cette forme qu'obéit le modèle apparaissant dans l'équation (34), avec

$$\underline{\beta}_{ab} = -m_a m_b \frac{16\nu}{\rho_a + \rho_b} \left|\nabla_a w_h\left(r_{ab}\right)\right| \frac{\underline{r}_{ab}}{r_{ab}^3} \tag{55}$$

(qui est bien antisymétrique). La conservation du moment cinétique résulte alors du fait que ces forces sont alignées avec $\underline{r}_{ab}$; or le comportement des fluides visqueux newtoniens nous enseigne qu'au contraire cette force doit être dirigée selon $\underline{u}_{ab}$, ce qui s'obtient en considérant un tenseur $\underline{\underline{\alpha}}_{ab}$ sphérique, c'est-à-dire se ramenant à un scalaire α_{ab} :

$$\underline{F}_{\nu,a} = -\sum_b \alpha_{ab} \underline{u}_{ab} \tag{56}$$

Cette forme correspond alors au modèle de l'équation (33), avec

$$\alpha_{ab} = -m_a m_b \nu \frac{\rho_a + \rho_b}{\rho_a \rho_b} \frac{\underline{r}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right)}{r_{ab}^2} \tag{57}$$

Cette forme ne conserve plus le moment. Nous constatons ainsi que les deux modèles possèdent leurs avantages propres : l'un (34) conserve le moment et s'annule pour une rotation solide, l'autre présente une forme de cisaillement conforme à l'intuition (voir **figure 6**). L'expérience numérique semble montrer que ce dernier est plus précis (Issa, 2005), tandis que le premier assure une meilleure stabilité numérique. Tous deux conservent en revanche l'impulsion, comme nous l'avons vu précédemment.

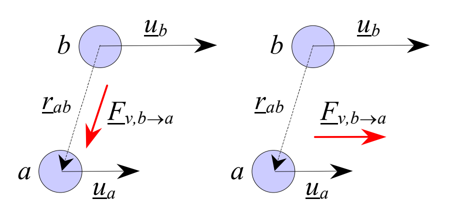


Figure 6 – Les deux formes de forces visqueuses dans le formalisme SPH : à gauche, la forme (34) conservant le moment, à droite la forme (33) se présentant comme un cisaillement.

Il est impossible, à partir de la forme générale (51), de combiner tous les avantages indiqués ici ; il faudrait pour cela un modèle consistant à associer à chaque particule un tenseur taux de déformation estimé en dérivant les vitesses grâce à (20), dont on calculerait la divergence grâce à (22) (pour des raisons de symétrie préservant alors la loi d'action-réaction). Bien que certains auteurs utilisent cette approche avec succès, elle n'a pas été retenue dans la version 1.2 de SPARTACUS-2D. Nous reviendrons pourtant brièvement sur ces considérations lorsque nous traiterons la modélisation de la turbulence (Cf. § **1.1.3.5**).

1.1.2.5 Equation d'état

Le problème se pose, pour un modèle lagrangien, d'empêcher les particules de se rapprocher indéfiniment, ce qui aurait pour effet d'augmenter localement la densité. Il est en particulier difficile de modéliser un écoulement incompressible. Une solution fréquemment utilisée pour SPH, et retenue dans SPARTACUS-2D, consiste à établir une équation d'état pour le calcul de la pression (qui demeure inconnue pour l'instant). Cette équation, pour l'eau, se présente sous la forme suivante :

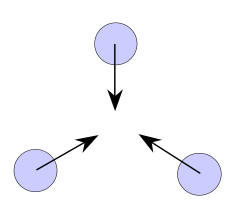


Figure 7 – Cas de trois particules se rapprochant.

$$p_a = \frac{\rho_0 c_0^2}{\gamma}\left[\left(\frac{\rho_a}{\rho_0}\right)^\gamma - 1\right] \qquad (58)$$

où ρ_0 représente une densité de référence, c_0 une vitesse du son numérique[9] au sein du fluide au repos (données par l'utilisateur), et $\gamma = 7$. Du fait de la valeur élevée de c_0 (ce qui correspond à un fluide très peu compressible), une faible variation locale de densité se traduit par une forte variation de pression. Envisageons en effet le cas illustré par la **figure 7**, où plusieurs particules se rapprochent. Le champ de vitesse étant convergent au point occupant le centre, une intégration sur un petit volume entourant ce point montre que div $\underline{u} < 0$ en ce point. L'équation (26) montre alors que la densité y doit croître, ce qui est conforme à l'intuition. Un fort gradient de pression va alors apparaître grâce à l'équation (58), dirigé vers l'extérieur. Ce gradient, constituant une force de rappel, agira dans l'équation de mouvement (33) ou (34) pour empêcher « l'effondrement » des particules, qui conserveront alors un espacement presque constant au cours du temps.

Nous voyons aussi que la pression s'annule dès que la densité prend sa valeur de référence, ce qui permet de respecter naturellement une condition de pression nulle au niveau d'une surface libre.

L'équation (58) constitue la dernière équation du modèle SPH « standard » (c'est-à-dire sans modélisation des effets turbulents). Elle est utilisée pour évaluer la pression de chaque particule. On voit qu'il s'agit, par cette méthode, de modéliser un écoulement *quasi-incompressible*, et non réellement incompressible. Cela peut sembler plus réaliste encore qu'un modèle incompressible, puisque tout se passe ici comme dans un fluide réel. Nous verrons cependant au **§ 1.2.1.1** que tel n'est pas le cas, pour des raisons numériques. Il est important de noter que la modélisation d'un fluide sensiblement *compressible*, parfaitement envisageable par cette méthode, exigerait néanmoins l'ajout de termes visqueux supplémentaires dans l'équation de quantité de mouvement (32) (Cf. note de bas de page du **§ 1.1.2.2**).

L'utilisateur a, bien entendu, la possibilité de modifier l'équation d'état s'il le souhaite (en particulier si plusieurs fluides sont modélisés), à condition de respecter une approche quasi-incompressible. Le choix de la vitesse du son est cependant sujet à des conditions numériques explicitées au **§ 1.2.1.1**.

Malgré les avantages de la description qui précède, l'approche retenue ici est loin d'être parfaite. De nombreux auteurs lui reprochent de donner lieu à des instabilités, en découplant la vitesse et la

[9] Comme nous le verrons au **§ 1.2.1.4**, cette grandeur diffère en principe de la vitesse du son réelle.

pression (voir par exemple Shao et Gotoh, 2004). Des méthodes existent, fondées sur la résolution d'une équation de Poisson pour la pression, qui permettent de s'affranchir de ces problèmes récurrents. La technique présentée ici, retenue pour SPARTACUS-2D V1P2, possède cependant l'avantage de la simplicité.

1.1.3 Modélisation de la turbulence

1.1.3.1 Principe

Nous allons d'abord résumer quelques généralités sur la turbulence, afin de poser le problème et les notations, et de donner les solutions usuelles à ce problème. Pour plus de détails, on pourra se référer à Viollet *et al.* (1998) ou Violeau (2004*a*).

Le nombre de Reynolds d'un écoulement, rapport Re = UL / ν (U et L étant des échelles caractéristiques de vitesse et d'espace), caractérise l'ordre de grandeur du rapport des forces d'inertie (convection de la vitesse) aux forces visqueuses. Lorsque ce nombre dépasse un seuil de l'ordre de 2000 (condition très fréquente), l'écoulement passe d'un régime dit *laminaire*, ordonné et stable, à un régime dit *turbulent*, c'est-à-dire chaotique : instationnaire, extrêmement instable, géométriquement complexe. La petitesse de certains tourbillons, ainsi que leur aspect résolument tridimensionnel, interdisent alors leur reproduction numérique exacte à moins de disposer d'une résolution spatiale et temporelle hors de proportion avec les possibilités actuelles des machines pour la prédiction d'un écoulement réel. D'autre part, le caractère extrêmement sensible de ces structures leur enlève toute reproductibilité, et donc une partie de leur signification physique. Pourtant, ces tourbillons ont un effet tel sur l'écoulement dans son ensemble qu'il est impensable de les oublier purement et simplement.

L'approche la plus populaire (Viollet *et al.*, 1998) consiste alors à considérer des grandeurs (vitesse et pression) moyennes au sens dit « de Reynolds », qui résultent d'un filtrage des grandeurs réelles. Les équations de Navier-Stokes (26) et (32) aboutissent alors aux équations de Reynolds :

$$\frac{d\overline{\rho}}{dt} = -\overline{\rho}\nabla \cdot \underline{\overline{u}} - \nabla \cdot \underline{R} \qquad (59)$$

$$\frac{d\underline{\overline{u}}}{dt} = -\frac{1}{\rho}\nabla \overline{p} + \nu\Delta\underline{\overline{u}} - \frac{1}{\rho}\nabla \cdot \left(\rho\underline{\underline{R}}\right) + \underline{F}^e \qquad (60)$$

qui sont en tous points analogues aux précédentes[10], à l'exception de deux termes additionnels venus perturber les membres de droite, faisant intervenir deux nouvelles grandeurs, l'une vectorielle, l'autre tensorielle :

$$\underline{R} = \overline{\rho' \underline{u}'} \qquad (61)$$

[10] Nous devrions préciser que la dérivée matérielle apparaissant à gauche dans les équations (59) et (60) représente désormais une dérivée « en suivant le mouvement moyen », ce qui en principe pose un certain nombre de questions de légitimité, car il est douteux qu'on puisse faire bouger une particule avec une vitesse n'ayant pas de réalité autre que statistique. Il est possible de contourner ce problème en admettant que des particules puissent échanger de la matière entre deux instants. Ce phénomène est du reste au fondement d'un comportement visqueux, ce qui est cohérent avec les modèles de fermeture adoptés dans la suite. Ce qui vient d'être dit concerne bien entendu toutes les équations apparaissant dans la suite et contenant une dérivée totale, c'est-à-dire (82), (83) et (92).

$$\underline{\underline{R}} = \overline{\underline{u}' \otimes \underline{u}'} \quad (62)$$

Les primes désignent les fluctuations des quantités concernées, c'est-à-dire les différences entre grandeurs réelles et grandeurs moyennes, si bien que les deux nouvelles quantités définies par (61) et (62) représentent des corrélations statistiques. Précisons dès à présent que la densité fluctue uniquement pour un fluide compressible, ce qui sera *a priori* le cas pour les fluides modélisés par SPARTACUS-2D puisque nous travaillons en régime quasi-incompressible. Néanmoins, les fluctuations de densité demeurent très faibles en raison de la valeur élevée de la vitesse du son apparaissant dans l'équation d'état (58) (Cf. **§ 1.2.1.4** pour plus de précisions à ce sujet). Aussi, nous considérerons que ces fluctuations sont négligeables, ce qui revient à poser $\overline{\rho} = \rho$ et $\underline{R} = \underline{0}$. Ainsi, l'équation de continuité moyennée demeure identique à l'équation initiale de Navier-Stokes, à ceci près qu'elle est fondée sur le champ de vitesse moyen :

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \overline{\underline{u}} \quad (63)$$

Il reste donc à traiter le terme en $\underline{R}$, dont l'approximation qui précède permet de dire qu'il se ramène à $-\nabla \cdot \underline{\underline{R}}$. Le tenseur $\underline{R}$, nommé *tenseur des contraintes de Reynolds*, représente l'effet des tourbillons sur l'écoulement moyen, et résulte du terme non linéaire de l'équation de quantité de mouvement de Navier-Stokes, implicitement inclus dans le terme d'accélération du membre de gauche (inertie). Ce tenseur demeure une inconnue du problème, et requiert donc l'intervention d'un modèle spécifique. Or, l'intuition et l'observation permettent de se rendre compte du double effet de ces fluctuations : elles exercent d'une part un mélange (diffusion), et d'autre part accroissent le taux de dissipation d'énergie en augmentant localement le frottement dans des proportions considérables. Ainsi, par analogie avec la loi de comportement d'un fluide visqueux, un modèle fondé sur une hypothèse de viscosité semble ici approprié :

$$\underline{\underline{R}} = \frac{2}{3} k \underline{\underline{I}} - 2\nu_T \underline{\underline{S}} \quad (64)$$

où $\underline{\underline{I}}$ est le tenseur identité et $\underline{\underline{S}}$ le tenseur taux de déformation moyen, défini comme la partie symétrique du gradient des vitesses moyennes :

$$\underline{\underline{S}} = \frac{1}{2}\left(\nabla \overline{\underline{u}} + {}^T \nabla \overline{\underline{u}}\right) \quad (65)$$

La quantité k, définie par

$$k = \frac{1}{2} \operatorname{tr} \underline{\underline{R}} \quad (66)$$

représente l'*énergie cinétique* des tourbillons, et dépend des variables de temps et d'espace. Elle demeure une inconnue du problème, tout comme la *viscosité turbulente* ν_T. Le modèle (64), introduit à la fin du XIXe siècle par Boussinesq, est certes très simple[11] mais efficace pour traiter un grand nombre d'écoulements. Il se décline en une famille de modèles pour estimer ν_T, dont nous examinerons aux paragraphes suivants les trois plus populaires, programmés dans SPARTACUS-2D.

---

[11] Voir à ce sujet la note du bas de la page 34.

Terminons cette introduction en donnant finalement l'équation de quantité de mouvement résultant de la combinaison de (60) et (64) :

$$\frac{d\underline{\bar{u}}}{dt} = -\frac{1}{\rho}\nabla \bar{p} + \nabla \cdot \left[2(\nu + \nu_T)\underline{\underline{S}}\right] + \underline{F}^e \tag{67}$$

On remarquera que le terme de (64) dépendant de k a disparu. Il peut être en effet introduit dans le terme de pression, donnant lieu à une redéfinition de cette dernière grandeur ; du reste il demeure très souvent négligeable devant la pression proprement dite ; dans SPARTACUS-2D V1P2, on considère cependant que dans le gradient de pression, il est nécessaire d'ajouter $2\rho k / 3$ à $\bar{p}$, à condition de connaître k (voir **§§ 1.1.3.3** et **1.1.3.4**). Excepté cette subtilité, (67) montre que tout revient à ajouter à la viscosité moléculaire la viscosité turbulente. Cette dernière lui est généralement très supérieure[12] ; il est cependant plus prudent de conserver ν dans l'équation (67) pour des raisons numériques. Dans la suite, cependant, pour plus de simplicité, nous remplacerons implicitement $\nu + \nu_T$ par ν_T dans tous nos calculs littéraux. Ajoutons que le modèle présenté ici englobe le cas laminaire comme un cas particulier pour lequel la viscosité turbulente serait nulle, les grandeurs moyennes se ramenant alors à de vraies grandeurs.

Les particules seront donc désormais affectées de vitesses et pressions moyennes. La forme adoptée pour les équations (63) et (67) dans SPARTACUS-2D est bien entendu analogue à (31), (35) et (36) :

$$\frac{d\rho_a}{dt} = \sum_b m_b \underline{\bar{u}}_{ab} \cdot \nabla_a w_h(r_{ab}) \tag{68}$$

$$\frac{d\underline{\bar{u}}_a}{dt} = \sum_b m_b \left(\underline{\bar{q}}_{p,ba} + \underline{\bar{q}}_{\nu,ba}\right) + \underline{F}_a^e \tag{69}$$

$$\frac{d\underline{r}_a}{dt} = \underline{\bar{u}}_a \tag{70}$$

avec des notations évidentes, et les choix suivants pour les flux de pression :

$$\underline{\bar{q}}_{p,ba} = -\left(\frac{\bar{p}_a}{\rho_a^2} + \frac{\bar{p}_b}{\rho_b^2}\right)\nabla_a w_h(r_{ab}) \tag{71}$$

$$\underline{\bar{q}}_{p,ba} = -\left(\frac{\bar{p}_a + \bar{p}_b}{\rho_a \rho_b}\right)\nabla_a w_h(r_{ab}) \tag{72}$$

et pour les flux visqueux :

$$\underline{\bar{q}}_{\nu,ba} = 8\frac{\nu_{T,a} + \nu_{T,b}}{\rho_a + \rho_b}\frac{\underline{\bar{u}}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2}\nabla_a w_h(r_{ab}) \tag{73}$$

$$\underline{\bar{q}}_{\nu,ba} = \frac{\rho_a \nu_{T,a} + \rho_b \nu_{T,b}}{\rho_a \rho_b}\frac{\underline{\bar{u}}_{ab}}{r_{ab}^2}\underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) \tag{74}$$

---

[12] Sauf dans les régions très proches d'une paroi lisse (présence d'une sous-couche visqueuse), qui sortent du contexte des applications de SPARTACUS-2D, les bords solides étant toujours rugueux en nature (Cf. à ce sujet le **§ 1.1.4.4**).

conformément à ce qui a déjà été présenté au **§ 1.1.2.2** dans le cas d'un écoulement laminaire. Quant à la pression moyenne, son calcul demeure inchangé par rapport au cas laminaire :

$$\overline{p}_a = \frac{\rho_0 c_0^2}{\gamma}\left[\left(\frac{\rho_a}{\rho_0}\right)^{\gamma} - 1\right] \quad (75)$$

Le choix d'un modèle de turbulence dans SPARTACUS-2D consiste donc à choisir une méthode de calcul de ν_T parmi les trois proposées aux paragraphes suivants.

1.1.3.2 Modèle de longueur de mélange

La méthode la plus simple pour estimer la viscosité turbulente est le fameux modèle de longueur de mélange introduit par Prandtl en 1925. Il s'agit tout d'abord de relier ν_T à k en faisant appel à l'analyse dimensionnelle (Viollet *et al.*, 1998) :

$$\nu_T = C_\mu \frac{k^2}{\varepsilon} \quad (76)$$

où ε est le *taux de dissipation* d'énergie cinétique turbulente (Cf. **§ 1.1.3.3** et notamment équation (82)). C_μ est une constante sans dimension estimée empiriquement à 0,09. Un autre raisonnement dimensionnel nous permettrait également d'écrire

$$\nu_T = C_\mu^{1/4} L_m k^{1/2} \quad (77)$$

où L_m est une longueur de mélange caractérisant l'échelle de diffusion exercée par les grands tourbillons, ce qui constitue une formulation analogue à celle de la viscosité moléculaire en cinétique des gaz. Enfin, l'hypothèse d'un équilibre entre production P et dissipation d'énergie ε (Cf. **§ 1.1.3.3**) aboutit à

$$\varepsilon \approx P \quad (78)$$

A partir du modèle de Boussinesq introduit au **§ 1.1.3.1**, il est possible d'écrire le taux de production d'énergie sous la forme

$$P = \nu_T S^2 \quad (79)$$

où S est le taux le déformation scalaire, défini par $S = (2S_{ij}S_{ij})^{1/2} = (2\underline{\underline{S}} : \underline{\underline{S}})^{1/2}$ (avec une somme sur i et j – convention d'Einstein), le symbole « : » désignant le double produit contracté de deux tenseurs. Les quatre relations qui précèdent donnent finalement

$$\nu_T = L_m^2 S \quad (80)$$

Ce modèle possède l'avantage de la simplicité, car il donne directement ν_T et n'exige le calcul que d'un seul paramètre, à savoir le taux de déformation S, calcul qui sera abordé au **§ 1.1.3.5**. En revanche, il est fondé sur un trop grand nombre d'hypothèses (en particulier l'équilibre (78)) pour être valable hors d'un petit nombre de cas simples, essentiellement en régime permanent et pour des écoulements cisaillés. Les écoulements aux fortes déformations ou rapidement changeants (tel l'écoulement au voisinage d'une marche descendante) ne peuvent être reproduits avec précision à l'aide de (80). Dans SPARTACUS-2D, le choix de ce modèle donne pour chaque particule

$$\nu_{T,a} = L_{m,a}^2 S_a \quad (81)$$

Ce modèle nécessite la donnée de L_m, constante dans le temps mais dépendant de l'espace et souvent inconnue pour une géométrie arbitraire (Violeau, 2004*a*). Il est toujours possible de choisir L_m constante, égal par exemple à la taille des particules (c'est ce qui est fait par défaut dans SPARTACUS-2D), mais alors il s'agit d'un paramètre numérique (et non plus physique), et l'estimation des profils de vitesse, notamment près des parois solides, s'en trouve altérée[13]. Le cas d'un écoulement pariétal (éventuellement à surface libre) est d'ailleurs pratiquement le seul où la longueur de mélange soit connue, proportionnelle à la distance à la paroi (on trouvera un exemple d'application dans la 3e partie, **§ 3.2**).

1.1.3.3 Modèle k–L_m

L'inconvénient essentiel du modèle de longueur de mélange est d'être fondé sur un équilibre erroné entre production et dissipation d'énergie (78). A partir de l'équation de Reynolds (67), il est en effet possible de montrer que l'énergie cinétique turbulente k obéit à l'équation suivante (Viollet *et al.*, 1998) :

$$\frac{dk}{dt} = \nabla \cdot \left(\nu + \frac{\nu_T}{\sigma_k} \nabla k \right) + P - \varepsilon \tag{82}$$

où σ_k est estimé égal à 1,0. P représente le taux de production d'énergie cinétique turbulente dû aux gradients de vitesse moyenne, donné par (79).

Nous voyons alors que l'équilibre (78) qui avait permis d'établir le modèle de longueur de mélange résulte directement de (82) à condition de supposer nuls les termes de variation dans le temps et de diffusion (turbulence permanente et homogène), ce qui est bien entendu très grossier. En revanche, l'équation (82) permet une meilleure estimation de k si l'on peut la résoudre. Pour cela, commençons par observer qu'il s'agit de l'équation de transport-diffusion d'un scalaire (compté *par unité de masse*), la production et la dissipation jouant le rôle de termes sources. La forme la plus générale d'une telle équation est

$$\frac{dC}{dt} = \nabla \cdot \left(\nu + \frac{\nu_T}{\sigma_C} \nabla C \right) + S_C \tag{83}$$

où σ_C est le nombre de Schmidt associé à la diffusion turbulente du scalaire C (Violeau, 2004*a*) et S_C un terme source. A partir de l'approximation (25), nous pouvons en écrire une forme SPH :

$$\frac{dC_a}{dt} = \sum_b m_b \frac{\rho_a \nu_{C,a} + \rho_b \nu_{C,b}}{\rho_a \rho_b} \frac{C_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) + S_{C,a} \tag{84}$$

C_a représentant la valeur du scalaire attachée à la particule a, tandis que $C_{ab} = C_a - C_b$. Chaque particule se voit attribuer une « diffusivité efficace pour C » définie par $\nu_C = \nu + \nu_T / \sigma_C$. Avant d'aller plus loin, il convient de préciser que le terme de diffusion apparaissant dans (84) est – comme l'était le terme visqueux diffusant les vitesses dans (33) et (34) – symétrique, ce qui possède une signification physique forte :

[13] On pourrait alors parler de « LES bidimensionnelle », bien que la simulation des grandes échelles (Large Eddy Simulation, ou LES) ne s'applique à strictement parler qu'en trois dimensions (Cf. par exemple Issa, 2005).

$$q_{C,ba} = m_b \frac{\rho_a \nu_{C,a} + \rho_b \nu_{C,b}}{\rho_a \rho_b} \frac{C_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) \tag{85}$$

représente en effet le flux de C de b vers a. Nous pouvons faire un raisonnement désormais classique : la symétrie de (85) signifie que ce qui est perdu par b en faveur de a est effectivement récupéré par a, donc qu'il y a conservation de la quantité

$$\sum_a m_a C_a \tag{86}$$

c'est-à-dire de la quantité totale du scalaire. C'est bien là la caractéristique d'un terme de diffusion, qui se ramène à un terme de flux à travers le bord lorsqu'on l'intègre sur un volume. Ainsi, la forme SPH des équations qui suivent satisfera de manière exacte ce principe, qui veut que la diffusion se contente de redistribuer la quantité concernée dans l'espace au cours du temps, au gré du coefficient de diffusion. L'apport ou la perte de cette substance est le fait du seul terme source S_C.

Nous parvenons alors à une forme SPH de l'équation permettant de calculer l'évolution de k pour chaque particule à chaque instant :

$$\frac{dk_a}{dt} = \sum_b m_b \frac{\rho_a \nu_{k,a} + \rho_b \nu_{k,b}}{\rho_a \rho_b} \frac{k_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) + P_a - \varepsilon_a \tag{87}$$

avec $k_{ab} = k_a - k_b$, $\nu_k = \nu + \nu_T / \sigma_k$ et

$$P_a = \nu_{T,a} S_a^2 \tag{88}$$

Pour fermer l'équation (82) (ou (87)), il est nécessaire d'estimer la dissipation, ce qui peut être fait en première approximation en considérant (76) et (77) :

$$\varepsilon = C_\mu^{3/4} \frac{k^{3/2}}{L_m} \tag{89}$$

soit, pour chaque particule :

$$\varepsilon_a = C_\mu^{3/4} \frac{k_a^{3/2}}{L_{m,a}} \tag{90}$$

En fin de compte, la viscosité turbulente est estimée grâce à (77), c'est-à-dire :

$$\nu_{T,a} = C_\mu^{1/4} L_{m,a} k_a^{1/2} \tag{91}$$

ce qui revient à utiliser la formule initiale (77), compte tenu de (91). Par rapport au modèle de longueur de mélange, le procédé présenté ici prend en compte le caractère instationnaire de la répartition de l'énergie. En particulier, l'énergie n'apparaît localement qu'en présence d'un terme de production suffisamment élevé, et avec un temps de retard de l'ordre de grandeur de la période des grands tourbillons. D'un autre côté, la notion de longueur de mélange subsiste, et fonde l'estimation du taux de dissipation (89), ce qui est clairement une approche simpliste.

L'équation (87), munie des modèles (90) et (91), est résolue dans SPARTACUS-2D à l'aide du schéma numérique présenté au **§ 1.2.1.2**. On parle alors de modèle à une équation ou modèle k–L_m. Encore une fois, il reste à calculer le taux de déformation S_a de chaque particule (Cf. **§ 1.1.3.5**) ; le

problème des conditions aux limites pour l'équation (87) sera abordé au **§ 1.1.4**.

1.1.3.4 Modèle k–ε

Le défaut de l'approche $k–L_m$, reposant sur une estimation grossière de la dissipation d'énergie et sur la donnée d'une longueur de mélange *ad hoc*, peut être levé avec le très populaire modèle k–ε. Il s'agit de conserver l'équation (82) pour calculer k, ε faisant en revanche l'objet d'un traitement analogue. On peut en effet construire une équation approchée sur ε, qui prend la forme suivante (Viollet *et al.*, 1998) :

$$\frac{d\varepsilon}{dt} = \nabla \cdot \left(\frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) + \frac{\varepsilon}{k}\left(C_{\varepsilon 1} P - C_{\varepsilon 2} \varepsilon \right) \tag{92}$$

Cette équation est formellement identique à (82), et peut donc être mise sous la forme SPH suivante :

$$\frac{d\varepsilon_a}{dt} = \sum_b m_b \frac{\rho_a \nu_{\varepsilon,a} + \rho_b \nu_{\varepsilon,b}}{\rho_a \rho_b} \frac{\varepsilon_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) + \frac{\varepsilon_a}{k_a}\left(C_{\varepsilon 1} P_a - C_{\varepsilon 2} \varepsilon_a \right) \tag{93}$$

avec des notations à présent évidentes. L'équation (93) prend la place de (90) pour l'estimation du taux de dissipation, la production étant toujours calculée grâce à (88). Enfin, la viscosité turbulente est calculée avec le modèle initial présenté au **§ 1.1.3.2** :

$$\nu_{T,a} = C_\mu \frac{k_a^2}{\varepsilon_a} \tag{94}$$

Là encore, nous aurons besoin d'un schéma numérique (**§ 1.2.1.2**) et de conditions aux bords (**§ 1.1.4**). Les constantes apparaissant dans (93) ont été affectées, dans SPARTACUS-2D, des valeurs recommandées par Launder et Spalding (1975) : σ_ε = 1,3 ; $C_{\varepsilon 1}$ = 1,44 ; $C_{\varepsilon 2}$ = 1,92.

Le modèle k–ε est extrêmement répandu dans les codes fondés sur des approches eulériennes, et a été présenté sous sa forme SPH pour la première fois par Violeau (2004*b*). Dans un contexte eulérien, il est connu pour donner de bons résultats dans de nombreux cas, et les validations réalisées grâce à SPARTACUS-2D sont également satisfaisantes (Cf. 3e partie, **§ 3.2**). La notion de longueur de mélange n'est plus nécessaire, ce qui en fait une méthode adaptée aux écoulements à géométrie complexe ; en revanche, par rapport au modèle de longueur de mélange, elle présente le désavantage d'exiger la résolution des deux équations supplémentaires (87) et (93). Le modèle $k–L_m$ présenté au paragraphe précédent constitue sans doute un compromis intéressant pour des écoulements pariétaux.

Un autre avantage des modèles $k–L_m$ et k–ε sur le modèle de longueur de mélange est la possibilité de prendre en compte l'énergie cinétique turbulente dans le terme de pression (voir la remarque à ce sujet en-dessous de l'équation (67)). Toutefois, ce terme supplémentaire ne prend de l'importance que pour les écoulement à nombre de Froude sensiblement supérieur à 10 (Violeau, 2004*a*).

Que l'on utilise le modèle $k–L_m$ ou k–ε, le terme de production doit être estimé. Nous avons donné en (88) le modèle courant. Guimet et Laurence (2002) ont cependant souligné le fait que cette approche surestime la production d'énergie (et donc k et ν_T) dans les zones de fortes déformations (au voisinage d'un point d'arrêt, par exemple), et proposé le modèle ci-dessous :

$$P_a = \min\left(\sqrt{C_\mu}, C_\mu S_a \frac{k_a}{\varepsilon_a}\right) k_a S_a \tag{95}$$

que nous écrivons ici directement sous sa forme SPH (*i.e.* avec des indices particulaires). Cette correction permet au modèle k–ε de fournir des résultats proches de ceux obtenus avec des modèles de turbulence au second ordre près d'un point d'arrêt[14], et est programmé dans SPARTACUS-2D. Que l'on estime la production avec (88) ou (95), nous voyons qu'il est indispensable, comme pour les deux autres modèles de turbulence présentés, de donner une estimation de S_a.

1.1.3.5 Calcul du taux de déformation

Le calcul du taux de déformation $S = (2\underline{\underline{S}} : \underline{\underline{S}})^{1/2}$ peut se faire de la manière suivante. On estime d'abord le tenseur gradient du champ de vitesse grâce à la formule d'interpolation (20), ce qui donne

$$\nabla\overline{\underline{u}}(\underline{r}_a) \approx -\frac{1}{\rho_a}\sum_b m_b \overline{\underline{u}}_{ab} \otimes \nabla_a w_h(r_{ab}) \tag{96}$$

puis on en déduit le tenseur taux de déformation à l'aide de sa définition (65), c'est-à-dire :

$$\underline{\underline{S}}_a = \frac{1}{2}\left(\nabla\overline{\underline{u}}(\underline{r}_a) + {}^T\nabla\overline{\underline{u}}(\underline{r}_a)\right) \tag{97}$$

Ensuite, on utilise tout simplement la définition de S donnée au **§ 1.3.2** :

$$S_a^2 = 2\underline{\underline{S}}_a : \underline{\underline{S}}_a \tag{98}$$

Cette approche possède cependant le désavantage de recourir 4 fois à la formule (20), pour ensuite utiliser des opérations supplémentaires ; elle n'est donc pas très efficace d'un point de vue algorithmique.

Une autre méthode consiste à revenir à la notion de dissipation. Nous avons vu au **§ 1.1.2.4** que la fonction F servant à l'établissement des forces visqueuses correspond en effet au taux de dissipation d'énergie (calcul (53)). Plus exactement, si E représente l'énergie *moyenne* du fluide, nous aurons

$$\frac{dE}{dt} = -\overline{F} \tag{99}$$

la barre supérieure faisant référence à une fonction de dissipation analogue à (51) mais fondée sur le champ de vitesse moyen. Or, en revenant à l'équation de Reynolds (60), il est possible de montrer que l'énergie perdue par le champ de vitesse moyen s'écrit exactement (Violeau, 2004*a*) :

$$\frac{dE}{dt} = -\int_\Omega \rho\nu_T S^2 d\Omega \tag{100}$$

---

[14] Il faut cependant préciser que l'approche k–ε présentée ici est encore loin d'être parfaite, notamment à cause de l'observation faite au bas de la page 24. La présence de fortes déformations doit en effet engendrer des termes de diffusion des vitesses qui n'apparaissent pas dans le modèle de Boussinesq (64), reposant sur des combinaisons non linéaires des composantes du tenseur taux de déformation (65). Les solutions usuellement proposées par la communauté scientifique pour prendre en compte ces phénomènes sont fondées essentiellement sur des modèles de fermeture au second ordre ou sur des modèles de viscosité non linéaires, précis mais complexes, et qui n'ont pas encore été mis en oeuvre dans SPARTACUS-2D à l'heure où ces lignes sont écrites ; on en trouvera cependant une brève description dans Violeau et Issa (2007).

Nous voyons du reste que cette énergie est intégralement récupérée par les tourbillons, puisque ce terme, changé de signe, est précisément la production d'énergie cinétique turbulente apparaissant dans (82). Ainsi, en revenant à la définition (51) de F et en utilisant l'approximation (10) pour estimer l'intégrale (100), nous avons

$$\frac{1}{2}\sum_{a,b} \underline{\bar{u}}_{ab}\,\underline{\underline{\alpha}}_{ab}\,\underline{\bar{u}}_{ab} = \bar{F} = \sum_a m_a \nu_{T,a} S_a^2 \qquad (101)$$

ce qui suggère

$$m_a \nu_{T,a} S_a^2 = \frac{1}{2}\sum_b \underline{\bar{u}}_{ab}\,\underline{\underline{\alpha}}_{ab}\,\underline{\bar{u}}_{ab} \qquad (102)$$

Revenant à présent à la définition des tenseurs $\underline{\underline{\alpha}}_{ab}$, nous aurons, avec la forme (57) moyennée (et en prenant en compte une viscosité turbulente, donc dépendante de la particule considérée) :

$$\nu_{T,a} S_a^2 = -\frac{1}{2}\sum_b m_b \frac{\rho_a \nu_{T,a} + \rho_b \nu_{T,b}}{\rho_a \rho_b} \frac{\bar{u}_{ab}^2}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) \qquad (103)$$

Nous proposons dans SPARTACUS-2D une forme simplifiée de cette équation, où l'on fait l'hypothèse que la viscosité turbulente est constante (uniquement dans l'équation (103)) :

$$S_a^2 \approx -\frac{1}{2}\sum_b m_b \frac{\rho_a + \rho_b}{\rho_a \rho_b} \frac{\bar{u}_{ab}^2}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) \qquad (104)$$

La positivité de cette grandeur est assurée par le fait que le gradient du noyau est dirigé de a vers b (comme nous l'avons déjà vu p. 14), tandis que $\underline{r}_{ab}$ est dirigé de b vers a. Par rapport à la méthode présentée plus haut, ce calcul du taux de déformation n'exige qu'une seule interpolation (au lieu de 4), et donne des résultats comparables (Cf. 3e partie, **§ 3.2.3**). En outre, il présente l'avantage d'être fondé sur le bilan d'énergie (100), ce qui assure au terme de production d'énergie, s'il est estimé par (88), de correspondre *exactement* à l'énergie perdue par l'écoulement moyen. Précisons que l'utilisation de (55) au lieu de (57) pour les $\underline{\underline{\alpha}}_{ab}$ donnerait une forme comparable à (104), mais sans apporter rien de plus, si bien que ce dernier choix n'a pas été retenu dans SPARTACUS-2D.

1.1.4 Conditions aux limites

1.1.4.1 *Frontières fluides*

Considérons d'abord le cas des surfaces libres. Un avantage de la méthode SPH est qu'aucune condition de pression n'y est requise, comme nous l'avons vu au **§ 1.1.2.5**. Aucune condition sur la vitesse n'est imposée dans SPARTACUS-2D, pas plus que pour les grandeurs turbulentes (k et ε). Pour ces grandeurs il est en effet d'usage de considérer des conditions de Neumann en surface (flux nul ou imposé), ce qui n'est pas aisé et demanderait des développements soignés, d'autant plus que la localisation des particules de surface est ambiguë. Cet inconvénient reste mineur, car l'expérience numérique a montré que les grandeurs physiques ont tendance à devenir naturellement homogènes au voisinage d'une surface libre, sans doute à cause de l'absence de particules au-delà de la surface. Le cas du taux de dissipation ε est cependant sujet à quelques interrogations, comme on le verra dans la 3e partie, au **§ 3.2**.

Pour une frontière liquide soumise à un flux sortant, aucune condition n'est requise, les particules qui traversent une telle frontière étant tout simplement retirées du lot.

Pour une frontière à flux entrant, en revanche, il est nécessaire d'imposer la position, la vitesse et la densité (dont est déduite la pression) de chacune des particules s'y trouvant à un instant donné, ainsi que les paramètres turbulents s'il y a lieu (k et ε), après quoi elles sont libres de se mouvoir comme les autres particules fluides. L'apparente simplicité de cette condition cache cependant une difficulté : il s'agit précisément de connaître l'allure du profil de ces paramètres le long d'une telle frontière, ce qui n'est pas toujours aisé. Il est aussi nécessaire d'imposer la forme complète de ces frontières, c'est-à-dire de connaître la hauteur d'eau lorsqu'il s'agit d'une frontière touchant une surface libre (Cf. 3e partie, **§ 3.2**).

Le cas d'une frontière « fictive » muni de conditions de périodicité ne nécessite aucun traitement particulier, sauf lors de la gestion des interactions entre particules ; ce cas est décrit au **§ 1.2.2.1**.

1.1.4.2 Parois solides : méthode des forces répulsives

La modélisation de parois solides doit être faite avec soin. L'approche la plus simple, retenue comme une option dans SPARTACUS-2D, consiste à placer le long de toute paroi solide une série de particules dites *de bord*, espacées de la valeur constante δr (distance initiale entre les particules fluides) modélisant la paroi proprement dite. En toute rigueur, elles représentent plutôt les particules de fluide qui adhèrent à ladite paroi (voir **figure 8**).

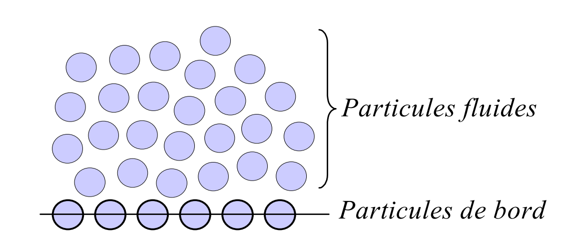


Figure 8 – Modélisation des parois par des particules de bord.

Chacune de ces particules possède une position et une vitesse déterminée au cours du temps (vitesse nulle dans le cas d'une paroi immobile, sauf en régime turbulent, comme on le verra au **§ 1.1.4.4**). Elle exerce une force répulsive sur les particules fluides qui s'en approchent suffisamment. La force exercée par la particule de bord b sur la particule fluide a prend la forme générale suivante :

$$\underline{F}^{p}_{b\to a} = \varphi(\tilde{q}_{ab})\frac{\underline{r}_{ab}}{r_{ab}} \tag{105}$$

où $\tilde{q}_{ab}$ est le rapport r_{ab} / r_0, r_0 étant la distance d'influence de la paroi. Dans SPARTACUS-2D, r_0 est égal à l'espacement initial entre particules δr, tandis que la fonction φ peut prendre l'une des trois formes ci-dessous :

- Forme de type purement répulsif (Monaghan, 1994) :

$$\varphi(\tilde{q}_{ab}) = \begin{cases} \dfrac{C}{r_0\tilde{q}_{ab}}\left(1/\tilde{q}_{ab}^{\;4} - 1/\tilde{q}_{ab}^{\;2}\right) & 0 \le \tilde{q}_{ab} \le 1 \\ 0 & 1 \le \tilde{q}_{ab} \end{cases} \tag{106}$$

  Cette force dépend d'une grandeur C ayant la dimension du carré d'une vitesse, fixée par l'utilisateur. Elle dépend de la configuration choisie : pour les problèmes tels que ruptures de barrages ou seuils (la paroi représentant le fond), avec une hauteur d'eau de l'ordre de H, il

est conseillé de choisir C de l'ordre de 5 à 10 fois gH. Ainsi, la force exercée par une paroi horizontale sur une particule située dans son voisinage est de l'ordre de grandeur de la force de pression exercée par la colonne d'eau sur cette même particule. Les particules fluides du fond conserveront donc un écartement à la paroi de l'ordre de δr .

- Forme de type « Lennard-Jones » (Monaghan, 1995) :

$$\varphi(\tilde{q}_{ab}) = \begin{cases} A(2/3\tilde{q}_{ab})^4 & 0 \le \tilde{q}_{ab} \le 2/3 \\ D(\tilde{q}_{ab} - 1)^2 & 2/3 \le \tilde{q}_{ab} \le 1 \\ 0 & 1 \le \tilde{q}_{ab} \end{cases} \qquad (107)$$

Au contraire de la précédente, cette force ne requiert la définition d'aucun paramètre par l'utilisateur, puisque les coefficients A et D sont définis par $A = 4c_0^2/81h$ et $D = 9 / A$ (rappelons que c_0 est la vitesse du son et h la longueur d'atténuation).

- Forme « avec cœur attractif » (Monaghan, 1995) :

$$\varphi(\tilde{q}_{ab}) = \begin{cases} \eta\left[(L/\tilde{q}_{ab})^m - (L/\tilde{q}_{ab})^n\right] & 0 \le \tilde{q}_{ab} \le 1 \\ A\left(\tilde{h} - \tilde{q}_{ab}\right)^2 + D\left(\tilde{h} - \tilde{q}_{ab}\right) & 1 \le \tilde{q}_{ab} \le \tilde{h} \\ 0 & \tilde{h} \le \tilde{q}_{ab} \end{cases} \qquad (108)$$

avec $(m,n) = (4,2)$ et $\tilde{h} = h / r_0$. Les quatre paramètres de cette loi sont définis par

$$\eta = \frac{2\sqrt{2}c_0^2}{27h} \qquad L = \left(\frac{n}{m}\right)^{\frac{1}{m-n}}$$
$$D = \frac{2\eta}{\tilde{h} - 1}\left(L^m - L^n\right) \qquad A = -\frac{D}{2\left(\tilde{h} - 1\right)} \qquad (109)$$

Quel que soit le modèle retenu, la force ne dépend donc que de la distance à la paroi. Nous observons qu'elle tend vers l'infini lorsque la particule fluide se rapproche d'une paroi, de manière à prévenir toute perméabilité. La force est en outre imposée nulle pour une particule s'éloignant suffisamment de la paroi ($r > r_0$). La force totale $\underline{F}_a^p$ ainsi exercée sur une particule fluide est ajoutée dans le membre de droite $\underline{F}_a^e$ de l'équation de quantité de mouvement (69). Ces forces, censées assurer l'imperméabilité de la paroi, sont fondées sur des modèles mathématiques moléculaires, ce qui leur donne un caractère un peu artificiel compte tenu de l'échelle macroscopique à laquelle on travaille généralement en mécanique des fluides. Il est donc recommandé d'avoir recours au modèle des particules fictives décrit au paragraphe suivant.

La méthode décrite ici fait apparaître un nouveau point fort de la méthode SPH, qui permet la prise en compte de parois mobiles : il suffit pour cela d'imposer aux particules de bord une position qui change au cours du temps. Il faut alors aussi leur affecter une vitesse, puisqu'elles sont prises en compte dans le calcul des forces visqueuses. Il est donc nécessaire de prendre en compte les particules de bord dans le second membre de l'équation de continuité (69).

1.1.4.3 Parois solides : méthode des particules fictives

La méthode des forces répulsives décrite plus haut fonctionne assez bien mais demeure un peu artificielle, et peut générer des instabilités locales. Il est donc préférable de recourir à une méthode plus consistante, dite des particules fictives : la paroi demeure matérialisée par des particules de bord, tandis que plusieurs couches de particules *fictives* se trouvent placées en-dessous et espacées de la même distance fixe δr, comme sur la **figure 9** (deux couches seulement y sont visibles, mais on en recommande plutôt 3 ou 4, comme on le verra).

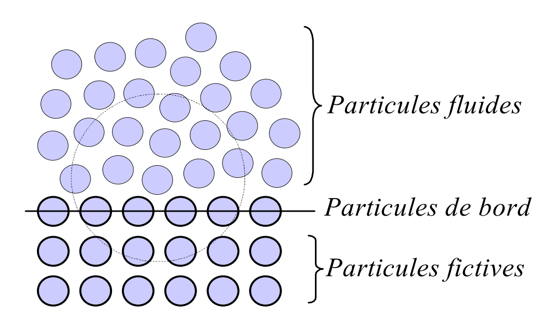


Figure 9 – Modélisation des parois par des particules fictives.

Les particules de bord n'exercent à présent plus de force répulsive artificielle, et l'imperméabilité de la paroi est modélisée par le procédé suivant. Au voisinage d'une paroi modélisée comme sur la **figure 8**, une particule de fluide possède des lacunes dans la liste des particules avec lesquelles elle interagit, tandis que le schéma de la **figure 9** permet, si les paramètres des particules fictives sont judicieusement réglés, de combler cette lacune (le nombre de couches de particules fictives doit donc être déterminé par le rapport h_t / δr – Cf. 2[e] partie, **§ 2.4.2.1**). On exploite alors la symétrie de l'équation de continuité (68) de manière à établir la densité des particules fictives : si une particule fictive b interagit avec une particule de fluide a, sa contribution à l'évolution de la densité de a sera

$$\frac{q_{V,ba}}{m_a} = m_b \underline{\overline{u}}_{ab} \cdot \nabla_a w_h (r_{ab}) \tag{110}$$

on incrémente alors la densité de b en lui ajoutant le même terme. De cette manière, la densité est symétrisée par rapport à la paroi ; la pression, calculée grâce à l'équation d'état (76), possèdera donc également cette propriété de symétrie, ce qui revient (en première approximation) à une condition de type Neumann sur la pression :

$$\left.\frac{\partial \overline{p}}{\partial \underline{n}}\right|_{paroi} = \underline{0} \tag{111}$$

où $\underline{n}$ est le vecteur normal à la paroi. Quant aux particules de bord, elles conservent une densité constante égale à la valeur de référence ρ_0. Cette condition s'est avérée nécessaire (dans le cas d'un écoulement gravitaire) pour éviter l'apparition d'une couche « de vide » près du bord (Issa 2005). Bien entendu, le procédé décrit ici exige que les particules fictives (ainsi que celles de bord) soient prises en compte dans les sommes définissant le flux de pression de l'équation (69) et le flux de volume de l'équation (68) ; il convient de préciser qu'elles sont également comptées dans le flux visqueux de (69). Sans cette précaution, en effet, l'expérience numérique montre que des irrégularités peuvent se former au voisinages des accidents géométriques des parois.

L'approche des particules fictives possède l'avantage de maintenir la propriété de conservation de l'impulsion totale du fluide vue au **§ 1.1.2.3**, tandis que l'approche du paragraphe précédent viole le respect rigoureux de cette loi. Ce qui a été dit au paragraphe précédent au sujet des parois mobiles demeure également vrai ici, à condition bien entendu de faire bouger les particules fictives avec la

même vitesse, de façon à garder une paroi solide. La méthode décrite ici possède donc de nombreuses qualités : signification physique, stabilité, précision (Cf. 3e partie, **§ 3.3**). En revanche, elle accroît légèrement le nombre de particules impliquées dans le calcul. Enfin, il convient de préciser que la plupart des auteurs (voir par exemple Morris *et al.*, 1997) envisagent des particules fictives mobiles (particules « miroirs »), symétriques des particules fluides réelles par rapport aux parois. Ce raffinement ne nous a pas paru utile, d'autant qu'il pose un problème pour le traitement des parois courbes.

Les avantages respectifs des deux méthodes (forces de paroi et particules fictives) seront brièvement examinés dans la 3e partie, au **§ 3.3.3**.

1.1.4.4 Traitement des grandeurs turbulentes aux parois

Nous allons enfin examiner le cas des grandeurs turbulentes, dont la prescription aux limites exige un point de vue légèrement différent, bien que fondé sur les mêmes outils que précédemment. Il faut en effet se souvenir qu'en régime turbulent le traitement des parois est délicat (Viollet *et al.*, 1998), soit parce qu'il existe dans son voisinage immédiat une *sous-couche visqueuse* très fine où les modèles de turbulence présentés plus haut échouent, soit parce que la rugosité de la paroi exerce une influence sur l'écoulement. On considère généralement, dans le contexte des approches de type « Reynolds » retenues pour SPARTACUS-2D, qu'une approche correcte est donnée par la notion de *loi de paroi*. Cette approche se révèle inévitable dans notre contexte, car la discrétisation particulaire interdit tout « raffinement » spécifique près des bords[15].

On ne peut donc désormais considérer que les particules de bord décrites plus haut sont réellement situées sur la paroi et y adhèrent, mais sont situées à une petite distance δ, plus grande que l'épaisseur de la sous-couche visqueuse ou que la taille des aspérités de la paroi. Compte tenu des valeurs typiques très faibles de ces deux derniers paramètres, il est néanmoins possible de choisir δ suffisamment petit devant δr, de sorte que les particules de bord pourront être *géométriquement* situées sur le bord, tout en ayant *physiquement* des caractéristiques correspondant à ce qui se passe à la distance δ de la paroi. En conclusion, la description des parois peut se faire alors de la même manière, mais les paramètres turbulents (vitesse moyenne, énergie cinétique et taux de dissipation) n'y seront pas nuls ; ils seront décrits par les lois de paroi suivantes (Viollet *et al.*, 1998) :

$$\bar{u} = u_* \left(\frac{1}{\kappa} \ln \frac{\delta}{k_s} + C_* \right) \tag{112}$$

$$k = \frac{u_*^2}{\sqrt{C_\mu}} \tag{113}$$

$$\frac{\partial \varepsilon}{\partial n} = -\frac{|u_*|^3}{\kappa \delta^2} \tag{114}$$

---

[15] Cette question peut être résolue par l'utilisation d'une longueur d'atténuation variable, mais nous pensons que ce concept est malaisé dans notre contexte, où les aspects convectifs dominent généralement. D'autre part, le recours à un h variable impose une modification de l'équation de quantité de mouvement présentée ici (Monaghan, 1992).

soit deux conditions de Dirichlet (vitesse et énergie) et une de Neumann (taux de dissipation[16]). Dans ces trois équations, $\kappa = 0{,}41$ est la constante de Karman, tandis que C_* vaut 8,5 ; k_s est l'épaisseur de rugosité de la paroi (constante en temps et en espace dans SPARTACUS-2D, mais fixée par l'utilisateur) et u_* la vitesse de frottement (au contraire susceptible de varier en temps et en espace), tandis que $\overline{u}$ représente la vitesse *tangente* à la paroi. Etant donné le champ d'application de SPARTACUS-2D (écoulements en nature), c'est donc une loi de paroi rugueuse qui a été retenue pour la vitesse, mais imposer une loi lisse ou mixte est aisé et a été réalisé avec succès (Issa, 2005). Imposer les conditions (112) et (113) se fait tout simplement en fixant la valeur de ces deux paramètres pour chaque particule de bord *a* :

$$\overline{\underline{u}}_a = u_{*,a}\left(\frac{1}{\kappa}\ln\frac{\delta}{k_s} + C_*\right)\underline{t}_a \tag{115}$$

$$k_a = \frac{u_{*,a}^2}{\sqrt{C_\mu}} \tag{116}$$

où $u_{*,a}$ est la vitesse de frottement locale correspondant à la particule de bord *a* (son procédé de calcul sera examiné plus loin) et où $\underline{t}_a$ désigne le vecteur tangent à la paroi en *a*, obtenu en faisant tourner d'un angle $-\pi / 2$ le vecteur normal $\underline{n}_a$ dirigé *vers l'intérieur* (**figure 10**). Il faut préciser que la vitesse imposée aux particules de bord par la loi (115) correspond à une grandeur physique susceptible de diffuser, mais ne doit en aucun cas servir au calcul du mouvement des parois, qui sont mues par l'utilisateur (s'il y a lieu) en vertu de la remarque faite à la fin du **§ 1.1.4.2**. D'autre part, et d'une manière générale, aucune condition n'est imposée aux particules fictives dans SPARTACUS-2D V1P2. Cela peut sembler entraver la diffusion des grandeurs cinématiques, mais demeure une solution aisée évitant le recours à des méthodes compliquées pour l'attribution de ces valeurs. Pour ces particules, nous aurons donc $\overline{u}$, k et $\varepsilon = 0$. La prescription de la condition de Neumann (114) est en revanche plus délicate (Violeau, 2004*b*). Nous avons vu au paragraphe précédent qu'il est aisé d'imposer une loi de flux nul à l'aide de la technique des particules fictives ; ici cependant nous sommes confrontés à deux difficultés : d'une part le second membre de (114) n'est pas nul, d'autre part ε varie généralement très rapidement près d'une paroi. L'expérience a montré qu'il est malcommode de donner aux particules fictives des valeurs non nulles de ε. Nous avons donc préféré imposer le flux (114) en fixant la valeur du terme de diffusion dans l'équation (93) portant sur ε. La loi (114) s'obtient en effet à partir de profils théoriques approchés de k et ε (Cf. 3[e] partie, **§ 3.2**), qui donnent en outre, au voisinage d'une paroi :

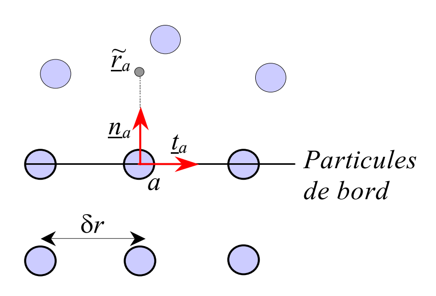


Figure 10 – Description des parois pour l'estimation des vitesses aux bords.

[16] Nous mettons la vitesse de frottement sous valeur absolue dans (114), car nous considérerons dans la suite qu'elle peut avoir une valeur négative, l'écoulement se faisant dans un sens ou dans l'autre.

$$\nabla \cdot \left(\frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{u_*^4}{\sigma_\varepsilon \delta^2} \tag{117}$$

La condition équivalente, dans SPARTACUS-2D, s'écrit donc pour chaque particule de bord :

$$\sum_b m_b \frac{\rho_a \nu_{T,a} + \rho_b \nu_{T,b}}{\sigma_\varepsilon \rho_a \rho_b} \frac{\varepsilon_{ab}}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) \leftarrow \frac{u_{*,a}^4}{\sigma_\varepsilon \delta^2} \tag{118}$$

la flèche signifiant que l'on remplace le membre de gauche par celui de droite. L'équation (93) est alors simplement résolue pour ces particules comme pour les particules fluides mais avec la forme (118) pour terme de diffusion, afin de déterminer ε_a aux bords. L'inconvénient majeur de cette solution est qu'elle viole le principe de conservation du scalaire ε, telle que nous l'avons exposée au **§ 1.1.3.3**. D'autre part, il peut arriver, dans des circonstances non stationnaires avec fortes déformations, que la condition (114) donne de trop faibles valeurs de ε, générant alors (*via* (94)) de trop fortes valeurs de viscosité turbulente. Pour cette raison, la condition suivante a été ajoutée pour les particules de bord dans SPARTACUS-2D :

$$\varepsilon_a \leftarrow \max\left(\varepsilon_a, \frac{\left| u_{*,a} \right|^3}{\kappa \delta} \right) \tag{119}$$

Une condition de Dirichlet seule sur ε ne donnerait en revanche pas de bons résultats, surestimant sa valeur au bord. Il est clair qu'un traitement adéquat de cette condition nécessitera dans l'avenir un travail plus approfondi.

Il nous reste à estimer la vitesse de frottement. Celle-ci, comme nous l'avons vu plus haut, dépend de la particule de bord considérée, et peut être déterminée de deux manières dans SPARTACUS-2D. La première méthode est fondée sur l'estimation de la vitesse près de la paroi ; le profil logarithmique (112) donne en effet une estimation de la vitesse moyenne sous la forme :

$$\overline{u}_{est} = u_* \left(\frac{1}{\kappa} \ln \frac{\delta_{est}}{k_s} + C_* \right) \tag{120}$$

à la distance δ_{est} d'une paroi. Si l'on parvenait à donner une autre estimation de cette vitesse, nous obtiendrions la vitesse de frottement en inversant cette loi :

$$u_{*,a} = \frac{\overline{u}_{est,a}}{\frac{1}{\kappa} \ln \frac{\delta_{est}}{k_s} + C_*} \tag{121}$$

où $\overline{u}_{est,a}$ représente la composante tangente de la vitesse estimée au point $\underline{\tilde{r}}_a$, situé à la distance δ_{est} d'une particule de bord *a* dans la direction normale à la paroi, comme en **figure 10**. Le problème est qu'il ne se trouve généralement pas de particule à cet endroit précis, mais l'interpolation SPH (9) nous permet d'écrire

$$\overline{u}_{est,a} \approx \sum_b \frac{m_b}{\rho_b} \underline{\overline{u}}_b \cdot \underline{t}_a w_h\left(\left| \underline{\tilde{r}}_a - \underline{r}_b \right| \right) \tag{122}$$

La distance d'estimation δ_{est} est fixée par l'utilisateur par l'intermédiaire de son rapport avec δr : α_{est} =

δ_{est} / δr (sur la **figure 15** ce rapport vaut 1, mais des tests numériques recommandent une valeur plus élevée, comme on le verra en 2e partie). Les formules (122) et (121) constituent une première méthode pour calculer les vitesses de frottement, un peu compliquée, il est vrai. Une autre méthode est fondée sur l'estimation du gradient de vitesse à la paroi, en utilisant la relation (96). On en déduit ensuite sa composante $_{xz}$, ou x représente l'axe orienté par $\underline{t}_a$, z celui orienté par $\underline{n}_a$ (**figure 10**) :

$$\left(\frac{\partial \bar{u}}{\partial n}\right)_a = \underline{t}_a \cdot \nabla \underline{\bar{u}}\left(\underline{r}_a\right) \cdot \underline{n}_a \qquad (123)$$

Enfin, on se fonde sur la loi logarithmique (112) pour en déduire le gradient de vitesse théorique :

$$\frac{\partial \bar{u}}{\partial n} = \frac{u_*}{\kappa \delta} \qquad (124)$$

en supposant qu'aux particules de bord on se trouve en réalité à la distance δ de la paroi, comme nous l'avons vu précédemment. Ainsi, (123) et (124) permettent d'écrire

$$u_{*,a} = \kappa \delta\, \underline{t}_a \cdot \nabla \underline{\bar{u}}\left(\underline{r}_a\right) \cdot \underline{n}_a \qquad (125)$$

Cette dernière méthode est recommandée pour sa simplicité. La première méthode citée, fondée sur la loi logarithmique du profil de vitesses, est cependant sans doute plus sûre.

1.2 ASPECTS NUMERIQUES ET ALGORITHMIQUES

1.2.1 Résolution des équations

1.2.1.1 Rappel des équations résolues

On rappelle ici les équations résolues par SPARTACUS-2D dans sa version 1.2. On se limitera au cas turbulent, le cas laminaire étant formellement identique avec des grandeurs moyennes coïncidant avec les grandeurs réelles, tandis que la viscosité turbulente est constante et égale à la viscosité cinématique moléculaire. Dans ce dernier cas, il faudrait également annuler les valeurs de vitesse de frottement apparaissant dans la condition (115).

Equation de continuité :

$$\frac{d\rho_a}{dt} = \sum_b m_b \underline{\bar{u}}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right) \qquad (126)$$

Equation de quantité de mouvement :

$$\frac{d\underline{\bar{u}}_a}{dt} = \sum_b m_b \left(\underline{\bar{q}}_{p,ba} + \underline{\bar{q}}_{\nu,ba}\right) + \underline{F}_a^e \qquad (127)$$

Equation du mouvement :

$$\frac{d\underline{r}_a}{dt} = \underline{\bar{u}}_a \qquad (128)$$

Equation d'état :

$$\overline{p}_a = \frac{\rho_0 c_0^2}{\gamma}\left[\left(\frac{\rho_a}{\rho_0}\right)^{\gamma} - 1\right] \quad (129)$$

Flux de pression :

Forme 1 $$\overline{\underline{q}}_{p,ba} = -\left(\frac{\overline{p}_a}{\rho_a^2} + \frac{\overline{p}_b}{\rho_b^2}\right)\nabla_a w_h\left(r_{ab}\right) \quad (130)$$

Forme 2 $$\overline{\underline{q}}_{p,ba} = -\left(\frac{\overline{p}_a + \overline{p}_b}{\rho_a \rho_b}\right)\nabla_a w_h\left(r_{ab}\right) \quad (131)$$

Flux visqueux :

Forme de Monaghan $$\overline{\underline{q}}_{\nu,ba} = 8\frac{\nu_{T,a} + \nu_{T,b}}{\rho_a + \rho_b}\frac{\overline{\underline{u}}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2}\nabla_a w_h\left(r_{ab}\right) \quad (132)$$

Forme de Morris $$\overline{\underline{q}}_{\nu,ba} = \frac{\rho_a \nu_{T,a} + \rho_b \nu_{T,b}}{\rho_a \rho_b}\frac{\overline{\underline{u}}_{ab}}{r_{ab}^2}\underline{r}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right) \quad (133)$$

Viscosité turbulente :

Modèle de longueur de mélange $$\nu_{T,a} = L_{m,a}^2 S_a \quad (134)$$

Modèles à une et deux équations $$\nu_{T,a} = C_\mu \frac{k_a^2}{\varepsilon_a} \quad (135)$$

$$\frac{dk_a}{dt} = \sum_b m_b \frac{\rho_a \nu_{k,a} + \rho_b \nu_{k,b}}{\rho_a \rho_b}\frac{k_{ab}}{r_{ab}^2}\underline{r}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right) + P_a - \varepsilon_a \quad (136)$$

Modèle k–L_m $$\varepsilon_a = C_\mu^{3/4}\frac{k_a^{3/2}}{L_{m,a}} \quad (137)$$

Modèle k–ε $$\frac{d\varepsilon_a}{dt} = \sum_b m_b \frac{\rho_a \nu_{\varepsilon,a} + \rho_b \nu_{\varepsilon,b}}{\rho_a \rho_b}\frac{\varepsilon_{ab}}{r_{ab}^2}\underline{r}_{ab} \cdot \nabla_a w_h\left(r_{ab}\right) + \frac{\varepsilon_a}{k_a}\left(C_{\varepsilon 1} P_a - C_{\varepsilon 2}\varepsilon_a\right) \quad (138)$$

Production d'énergie :

Modèle standard $$P_a = \nu_{T,a} S_a^2 \quad (139)$$

Modèle de Laurence $$P_a = \min\left(\sqrt{C_\mu}, C_\mu S_a \frac{k_a}{\varepsilon_a}\right)k_a S_a \quad (140)$$

Taux de déformation :

Modèle standard $$\nabla\overline{\underline{u}}\left(\underline{r}_a\right) \approx -\frac{1}{\rho_a}\sum_b m_b \overline{\underline{u}}_{ab} \otimes \nabla_a w_h\left(r_{ab}\right) \quad (141)$$

$$\underline{\underline{S}}_a = \frac{1}{2}\left(\nabla\overline{\underline{u}}\left(\underline{r}_a\right) + {}^T\nabla\overline{\underline{u}}\left(\underline{r}_a\right)\right) \quad (142)$$

$$S_a^2 = 2\underline{\underline{S}}_a : \underline{\underline{S}}_a \tag{143}$$

Modèle de Violeau

$$S_a^2 = -\frac{1}{2}\sum_b m_b \frac{\rho_a + \rho_b}{\rho_a \rho_b} \frac{\overline{u}_{ab}^2}{r_{ab}^2} \underline{r}_{ab} \cdot \nabla_a w_h(r_{ab}) \tag{144}$$

Ces équations sont sujettes aux conventions de notation résumées en **Annexe 1**, et munies des conditions aux limites données au **§ 1.1.4**.

1.2.1.2 Schéma numérique

Parmi les équations présentées ci-dessus, (126), (127) et (128) nécessitent un schéma numérique, ainsi que (136) et (138) le cas échéant. Un schéma de type Euler explicite d'ordre 1 a été retenu dans SPARTACUS-2D V1P2, pour des raisons de simplicité algorithmique et d'optimisation du temps de calcul. On introduit donc un pas de temps δt, et pour chaque particule a on réactualise les grandeurs $\underline{r}_a$, $\overline{\underline{u}}_a$, $\overline{p}_a$, ρ_a, k_a, ε_a, $\nu_{T,a}$, P_a et S_a à chaque pas de temps (noté d'un exposant n) à l'aide des opérations suivantes, effectuées *dans l'ordre indiqué* :

$$\begin{aligned}
&S_a^{n+1} = \psi_{S,1}\left(\underline{r}_b^n, \rho_b^n, \overline{\underline{u}}_b^n\right) && \text{ou} \quad S_a^{n+1} = \psi_{S,2}\left(\underline{r}_b^n, \rho_b^n, \overline{\underline{u}}_b^n\right)\\
&P_a^{n+1} = \psi_{P,1}\left(\nu_{T,b}^n, S_b^{n+1}\right) && \text{ou} \quad P_a^{n+1} = \psi_{P,2}\left(\varepsilon_b^n, k_b^n, S_b^{n+1}\right)\\
&k_a^{n+1} = k_a^n + \delta t \frac{dk_a}{dt}\left(\underline{r}_b^n, \rho_b^n, \nu_{T,b}^n, \varepsilon_b^n, k_b^n, P_b^{n+1}\right) &&\\
&\varepsilon_a^{n+1} = \varepsilon_a^n + \delta t \frac{d\varepsilon_a}{dt}\left(\underline{r}_b^n, \rho_b^n, \nu_{T,b}^n, \varepsilon_b^n, k_b^{n+1}, P_b^{n+1}\right) && \text{ou} \quad \varepsilon_a^{n+1} = \psi_\varepsilon\left(k_b^{n+1}, L_{m,b}\right)\\
&\nu_{T,a}^{n+1} = \psi_{\nu_T,1}\left(k_b^{n+1}, \varepsilon_b^{n+1}\right) && \text{ou} \quad \nu_{T,a}^{n+1} = \psi_{\nu_T,1}\left(S_b^{n+1}, L_{m,b}\right)\\
&\overline{\underline{u}}_a^{n+1} = \overline{\underline{u}}_a^n + \delta t \frac{d\overline{\underline{u}}_a}{dt}\left(\underline{r}_b^n, \overline{p}_b^n, \rho_b^n, \overline{\underline{u}}_b^n, \nu_{T,b}^{n+1}\right) &&\\
&\underline{r}_a^{n+1} = \underline{r}_a^n + \delta t \overline{\underline{u}}_a^{n+1} &&\\
&\rho_a^{n+1} = \rho_a^n + \delta t \frac{d\rho_a}{dt}\left(\underline{r}_b^n, \overline{\underline{u}}_b^{n+1}\right) &&\\
&\overline{p}_a^{n+1} = \psi_p\left(\rho_b^{n+1}\right) &&
\end{aligned} \tag{145}$$

Dans ces formules, les dérivées premières représentent les seconds membres de (126), (127), (128), (136) et (138), tandis que les fonctions $\psi_{S,1}$, $\psi_{S,2}$, $\psi_{P,1}$, $\psi_{P,2}$, ψ_ε, $\psi_{\nu T,1}$, $\psi_{\nu T,2}$ et ψ_p désignent respectivement les seconds membres de (144), (145), (139), (140), (138), (134), (135) et (129).

Il n'est pas du tout évident que l'ordre des opérations indiquées dans (145) soit le meilleur, mais la petitesse du pas de temps (Cf. **§ 1.2.1.4**) permet de penser que cet ordre est de peu d'importance ; cette question est laissée en suspens à l'heure actuelle, en attendant de plus amples investigations. De même, on peut s'interroger sur le choix d'un schéma numérique aussi simple[17] ; le modèle k–ε, par

[17] De nombreux auteurs recommandent des schémas plus complexes, de type Runge-Kutta ou prédicteur-correcteur, par exemple. Des essais dans SPARTACUS-2D ont cependant montré qu'ils n'apportaient rien dans certains cas (rupture de barrage et canal), sans doute grâce à l'emploi d'un très petit pas de temps (**§ 1.2.1.4**). D'ultérieurs développements permettront néanmoins de mettre en évidence les apports possibles de tels schémas sur la stabilité du code.

exemple, fait très souvent l'objet de schémas implicites pour traiter les termes de dissipation dans les codes eulériens, afin d'éviter l'apparition de valeurs négatives. Dans SPARTACUS-2D, il a été vérifié que cette approche n'apporte rien par rapport à l'approche complètement explicite présentée ici. La positivité des grandeurs k et ε est assurée par un *clipping* à 10^{-16} USI à chaque pas de temps (voir aussi à ce sujet le paragraphe suivant).

La donnée des paramètres de chaque particule à un instant donné permet donc de les connaître à l'instant ultérieur. La donnée de certains paramètres initiaux est évidemment nécessaire, à savoir leurs positions, vitesses et densités (données par l'utilisateur), ainsi que les valeurs de k et ε le cas échéant. Dans ce dernier cas, le code procède de la manière suivante. On peut estimer qu'un ordre de grandeur correct de l'énergie cinétique turbulente est le carré de la vitesse de frottement (Cf. par exemple l'équation (113)), elle-même de l'ordre de 0,05 fois la vitesse moyenne (Viollet *et al.*, 1998). Cette dernière étant majorée (en principe) par $u_{max} = 0{,}1c_0$ (voir **§ 1.2.1.4**), on estime k à l'instant initial par la grandeur constante $(0{,}4 \times 0{,}05 \times 0{,}1 \times c_0)^2$, le facteur 0,4 étant destiné à tempérer le choix $\overline{u} = u_{max}$:

$$k_a^0 = \left(0{,}002\, c_0\right)^2 \tag{146}$$

Quant à la dissipation initiale, elle est estimée grâce à (137) quel que soit le modèle de turbulence choisi. Dans le cas où l'utilisateur choisi le modèle k–ε, ne reposant sur aucune notion de longueur de mélange, le code utilise $L_{m,a} = \delta r$ pour cette initialisation :

$$\varepsilon_a^0 = C_\mu^{3/4} \frac{k_a^{3/2}}{\delta r} \tag{147}$$

Ces méthodes, utilisées dans la plupart des codes commerciaux classiques, peuvent aboutir à une estimation grossière des grandeurs turbulentes initiales, mais les équations (137) et (138) redonnent rapidement des valeurs physiques (avec un temps de relaxation de l'ordre de k / ε).

Les autres grandeurs initiales sont alors calculées à partir des fonctions ψ qui précédent.

1.2.1.3 Précautions numériques

Comme dans tout code, il est nécessaire de prendre un certain nombre de précautions concernant d'éventuelles divisions par zéro. Nous avons déjà évoqué, au **§ 1.1.1.2**, le fait que les dénominateurs des termes de diffusion sont minorés par un petit réel dans ce but précis ; il s'agit des équations (132), (133), (136), (138) et (144).

Un *clipping* semblable est imposé à l'équation (137), où $L_{m,a}$ est augmenté de la quantité $\eta = 0{,}1h$. D'autre part (comme nous l'avons vu au paragraphe précédent), après résolution de l'équation (138) la valeur de ε_a est minorée par 10^{-16} USI. Cette correction permet d'éviter les divisions par zéro dans les équations (135) et (140). Cette précaution est également prise dans l'initialisation des grandeurs turbulentes lorsqu'on procède à un calcul turbulent faisant suite à un autre calcul où les grandeurs turbulentes n'étaient pas calculées (**§ 2.4.2.7**).

Dans un autre registre, l'usage permet de noter que les champs de vitesses et surtout de pression, obtenus par l'algorithme décrit par les équations (145), sont souvent bruités. Cela provient essentiellement du caractère quasi-incompressible dudit algorithme, où le recours à une équation d'état rapidement non-linéaire telle que (129) amplifie les inévitables fluctuations numériques (de récents développements, non pris en compte dans la version 1.2 du code, ont montré que l'approche

réellement incompressible évoquée à la fin du **§ 1.1.2.5** permet de résoudre ce problème – voir à ce sujet Lee, 2005). Cependant, une alternative simple consiste à faire usage d'un procédé de lissage de la densité (Halem et Issa, 2006). Ce dernier consiste, après l'étape de continuité (126), à corriger la densité de chaque particule selon la formule

$$\rho_a^{n+1} \leftarrow \rho_a^{n+1} + 2\varepsilon_\rho \sum_b m_b \frac{\rho_a^{n+1} - \rho_b^{n+1}}{\rho_a^{n+1} + \rho_b^{n+1}} w_h^n\left(r_{ab}^n\right) \quad (148)$$

où ε_ρ est un petit paramètre sans unité, dont l'expérience numérique a montré qu'il doit être de l'ordre de grandeur de quelques centièmes. On voit que cela constitue une sorte de diffusion permettant une homogénéisation (et donc un lissage) de la densité, sans que cela ait trop d'effets physiquement artificiels puisque la densité est par nature presque constante. D'autre part, si la densité était réellement constante, l'équation (148) n'apporterait aucune modification. On note que (148) est la seule équation nécessitant la connaissance du noyau lui-même, et non de son gradient. Dans SPARTACUS-2D V1P2, compte tenu des résultats fiables donnés par le noyau d'ordre 4 donné par l'équation (7), c'est ce dernier qui a été retenu. Le procédé de lissage que nous venons de décrire s'avère très efficace et peu coûteux, mais demeure optionnel (Cf. **§ 2.2.1.3**). Un exemple est présenté en 3e partie, **§ 3.3.3**.

1.2.1.4 Pas de temps

Pour des raisons de stabilité numérique, le pas de temps doit vérifier trois contraintes :

$$\delta t = \min\left(0{,}25 \times \min_a \sqrt{\frac{h}{F_a}}\,;0{,}4 \times \frac{h}{c_0}\,;0{,}125 \times \min_a \frac{h^2}{\nu_{T,a}}\right) \quad (149)$$

Dans la première condition, on trouve les forces de masse F_a auxquelles les particules sont soumises (donc la norme du second membre de l'équation de quantité de mouvement dans (127)). Cette condition permet d'assurer que les particules ne s'approchent pas trop les unes des autres pendant l'intégration de leur mouvement sur un pas de temps. La seconde condition est une condition de CFL dépendant de la vitesse du son ; elle a pour objet de faire en sorte que les variations de pression ne se propagent pas d'une particule à une autre entre deux instants consécutifs, ce qui empêche l'apparition d'ondes de chocs artificielles. Enfin, la troisième condition est liée à la stabilité du terme visqueux, et est généralement prépondérante pour les écoulements à très faible nombre de Reynolds (Morris, 1997). Nous l'avons cependant retenue dans SPARTACUS-2D, afin de prévenir l'effet de trop grandes valeurs de la viscosité turbulente.

Les constantes apparaissant dans (149) ont été établies sur la base de tests numériques (Morris *et al.*, 1997). Nous voyons qu'il est nécessaire de réévaluer le pas de temps à chaque début de boucle, puisque sa valeur est censée varier en fonction des caractéristiques cinématiques de l'écoulement (premier et troisième termes de la formule (149)). Cependant, la deuxième condition étant généralement prépondérante, il arrive fréquemment que le pas de temps demeure constant.

Il faut à présent faire une remarque importante concernant la vitesse du son. Si l'utilisateur choisit de donner à c_0 sa vraie valeur (1400 ms^{-1} pour l'eau, à température et pression ordinaires), le pas de temps s'en trouve alors très petit, et les temps de calcul très élevés. Il est donc recommandé de donner à la vitesse du son une valeur artificielle, moins élevée :

$$c_0 = \frac{u_{\max}}{M_0} \qquad (150)$$

u_{max} étant la vitesse maximale au sein de l'écoulement (dont on connaît généralement une bonne estimation). M_0 est un nombre de Mach de référence, que l'on peut choisir égal à 0,1 si l'on veut modéliser un fluide quasi-incompressible. Une analyse dimensionnelle de l'équation de Navier-Stokes permet en effet de montrer que les variations relatives de densité dans un fluide compressible sont reliées au nombre de Mach par $\delta\rho / \rho \approx M^2$; la formule (150) assure alors des variations relatives de densité majorées par 1 %. On modélise ainsi un fluide artificiel, légèrement plus compressible que le fluide réel. Ce choix s'avère un bon compromis entre précision et rapidité de calcul. On notera cependant que Issa *et al*. (2004) recommandent, dans certains cas non gravitaires, de considérer $M_0 = 1 / 50$ pour éviter l'apparition de poches de vide en aval d'un obstacle ; la valeur de 0,1 n'est donc ici qu'indicative.

Il convient de noter qu'une condition supplémentaire existe pour les écoulements où le nombre de Froude est modéré (écoulements dominés par la gravité). Un profil de pression hydrostatique donne en effet, à l'aide de l'équation d'état (129), des variations relatives de densité de l'ordre de $\delta\rho / \rho \approx gH / c_0^2$, où H est la hauteur d'eau. Ainsi la condition (150) devient :

$$c_0 = \frac{1}{M_0} \max\left(u_{\max}, \sqrt{gH_{\max}}\right) \qquad (151)$$

ou $H_{\max}$ est la hauteur d'eau maximale. Notons dès à présent qu'il est peu commode (pour des raison de temps CPU), de reproduire par cette méthode des phénomènes où la vraie vitesse du son dans le fluide joue un rôle prépondérant. En particulier, les phénomènes liés au caractère faiblement compressible du fluide (coup de bélier, par exemple) nécessitent un choix de c_0 correspondant à sa vraie valeur, ce qui augmente considérablement le temps de calcul (d'un facteur M_0^{-1}, soit 10 en général). D'autre part, une telle approche rendrait caduque l'usage de l'équation d'état (129), qui devrait être remplacée par une équation de bilan d'énergie interne (Monaghan, 1992). Le choix de la vitesse du son est cependant laissé à l'utilisateur, les équations (150) et (151) n'étant pas programmées dans SPARTACUS-2D. Elles permettent simplement à l'utilisateur d'évaluer la valeur optimale de c_0 pour un écoulement où le caractère faiblement compressible du fluide ne joue aucun rôle, à condition d'avoir une idée des valeurs de $u_{\max}$ et $H_{\max}$.

1.2.2 Optimisation des calculs

1.2.2.1 Gestion des liens entre particules

Telles qu'elles ont été construites, les opérations effectuées par le programme (calcul des membres de droite des équations (126), (127), (136), (138), (141) et (144)) concernent chaque paire de particules, ce qui confère à l'algorithme un temps d'exécution en N^2, où N est le nombre de particules en jeu. On se souvient cependant que chaque particule n'interagit qu'avec un nombre fini de voisines ; il paraît donc opportun, à chaque pas de temps, de construire un tableau de liens entre particules, qui donne pour chacune la liste de celles avec lesquelles elle interagit. Le nombre d'opérations est ainsi proportionnel à N, et le temps calcul s'en trouve considérablement réduit. Pour cela, on procède de la manière suivante :

1 – On construit un quadrillage qui recouvre le domaine, et dont les cellules carrées ont un côté h_t (voir **figure 11**).

2 – Pour chaque particule, on détermine la cellule qui la contient.

3 – Pour chaque cellule, on détermine les particules qu'elle renferme.

4 – Pour chaque particule *a*, on considère les particules *b* se trouvant dans la même cellule ou dans l'une des huit cellules adjacentes, et l'on retient comme voisines celles qui satisfont à la condition $r_{ab} < h_t$.

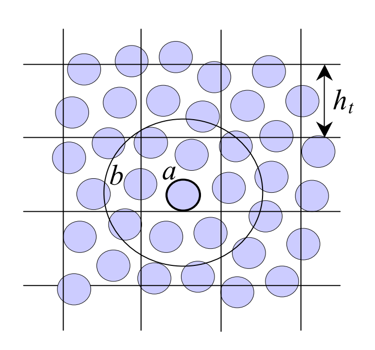


Figure 11 – Construction du tableau de liens.

Dans toutes les étapes de l'algorithme de résolution des équations, les termes faisant apparaître un signe « somme » ne sont donc calculés que pour des paires de particules liées. Si l'on souhaite simuler un écoulement périodique le long d'un certain axe (cas fréquent d'un canal, par exemple), pour lequel les particules sortant d'un côté réapparaissent aussitôt de l'autre avec les mêmes paramètres, il convient de faire en sorte que les particules situées près de l'un de ces bords soient liées à celles se trouvant près de l'autre (voir **figure 12**). Nous en trouverons un exemple en 3e partie (**§§ 3.2** et **3.3**).

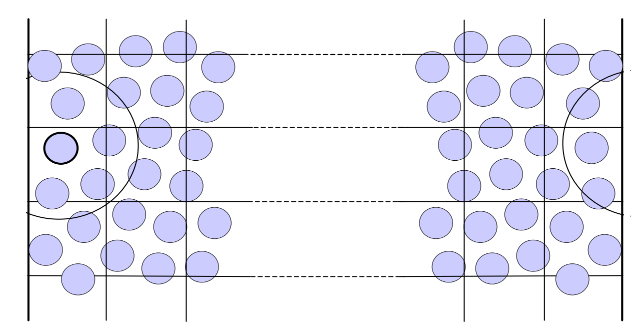


Figure 12 – Cas périodique horizontal.

1.2.2.2 Exploitation des symétries

Lors de la construction du tableau de liens, on ne comptabilise qu'une seule fois une paire de particules qui interagissent entre elles, en utilisant le principe symétrique qui veut que si une particule interagit avec une autre, cette dernière interagit automatiquement avec la première. On n'examine donc que les couples *a*, *b* tels que $a < b$. Ce procédé est également utilisé pour évaluer tous les termes des équations présentant une symétrie ou une antisymétrie, ce qui est toujours le cas grâce aux propriétés de symétrie du noyau. Outre le temps calcul, cette méthode permet aussi de réduire (d'un facteur 2) la mémoire requise. Par exemple, lors du calcul du terme de droite de l'équation de continuité (127), le second membre est symétrique. Pour évaluer l'évolution de la densité, on effectue alors les opérations suivantes pour toutes les particules *a* :

$$\begin{aligned} \forall b > a \quad X &= \underline{\bar{u}}_{ab} \nabla_a w_h(r_{ab}) \\ \rho_a &\leftarrow \rho_a + m_b X \delta t \\ \rho_b &\leftarrow \rho_b + m_a X \delta t \end{aligned} \tag{152}$$

Dans toute intervention de l'utilisateur visant à modifier les équations ou à en ajouter d'autres, il est important de ne pas oublier cette contrainte de programmation. Dans la version actuelle du code, elle concerne les équations (126), (127), (136), (138), (141) et (144).

1.2.3 Description algorithmique

1.2.3.1 Algorithme général de SPARTACUS-2D

Après une phase d'initialisation (positions et vitesses initiales des particules, grandeurs turbulentes initiales) et de définition des grandeurs constitutives des particules (type de fluide, par exemple) et des paramètres du calcul (densités de référence, viscosités moléculaires, vitesse du son, etc.), le calcul proprement dit peut commencer[18]. Compte tenu du paragraphe **1.2.1.2**, l'algorithme de SPARTACUS-2D se présente comme indiqué en **figure 13**.

Nous voyons que l'une des qualités du code (et de la méthode SPH en général) est son étonnante simplicité, d'un point de vue algorithmique. Le placement des particules, au premier pas de temps, peut se faire sur une grille régulière carrée, de côté δr, procédé recommandé (Cf. 3[e] partie).

1.2.3.2 Estimation du temps CPU

Le temps nécessaire à un calcul est déterminé en premier lieu par le nombre de particules, en second lieu par le choix du pas de temps (équation (149)). Or, ce dernier est directement proportionnel à la longueur d'atténuation h (en admettant que la condition de Courant domine), elle-même reliée à la distance interparticulaire δr. Pour une géométrie donnée, nous voyons que lorsque le nombre de particules N augmente, la distance entre particules δr diminue, et donc h aussi. Le pas de temps s'en trouve ainsi réduit, tandis que le nombre d'opérations augmente. Il est intéressant d'examiner la dépendance du temps de calcul en fonction de N, pour un phénomène dont les dimensions linéaires sont données, ainsi que le temps réel de déroulement $T_{\text{réel}}$. Evaluons pour cela le nombre N_{op} d'opérations effectuées à chaque pas de temps par l'algorithme :

$$N_{op} \propto NN_v = \pi(\alpha_t \alpha_h)^2 N \tag{153}$$

N_v étant le nombre de particules liées à une particule donnée, $\alpha_t = h_t / h$ le rapport entre le rayon du support du noyau et la longueur d'atténuation et $\alpha_h = h / \delta r$ le rapport entre cette dernière et la distance moyenne entre particules (Cf. **§ 1.1.1.1**). Ainsi, le temps total de calcul est donné par

$$T_{cpu} \propto (\alpha_t \alpha_h)^2 N \frac{T_{\text{réel}}}{\delta t} \tag{154}$$

Tenant compte de (149) (en supposant la deuxième condition majorante), nous avons donc

$$T_{cpu} \propto (\alpha_t \alpha_h)^2 N \frac{c_0 T_{\text{réel}}}{h} \tag{155}$$

Rappelons que la vitesse du son doit être choisie selon la condition (151). Si le nombre de particules est fixé (et suffisamment grand), leur écart initial (et donc h) est approximativement proportionnel à $H_{\max} / N^{1/2}$, où $H_{\max}$ est une dimension linéaire caractéristique du phénomène (hauteur d'eau maximale, par exemple). Ainsi :

$$T_{cpu} = C_T (\alpha_t \alpha_h)^2 N^{3/2} \frac{u_{\max} T_{\text{réel}}}{H_{\max}} \max(1, Fr) \tag{156}$$

---

[18] Des détails sur cette phase d'initialisation sont donnés dans la 2[e] partie.

où Fr est un nombre de Froude fondé sur la hauteur d'eau et la vitesse maximales. Le coefficient C_T (qui a la dimension d'un temps et sera déterminé dans la 3e partie, **§ 3.3.2**) dépend essentiellement des caractéristiques géométriques du cas considéré (indépendamment de l'échelle), notamment de la description des bords solides. On doit cependant considérer qu'il varie– dans une moindre mesure – en fonction du nombre d'équations résolues, c'est-à-dire principalement du modèle de turbulence choisi par l'utilisateur (voir **§ 3.5**), mais aussi du nombre de Mach numérique de référence M_0 (Cf. **§ 1.2.1.4**) et – cela va de soi – du type de machine utilisée. L'expérience numérique (voir 3e partie) montre que C_T est de l'ordre de grandeur de 2 à 3 fois 10^{-5} s.

Précisons pour terminer qu'il conviendrait en toute rigueur d'utiliser la formule (156) avec le nombre de particules fluides seules, car les autres sont sujettes à sensiblement moins d'opérations. Cette formule donne alors la dépendance du temps CPU en fonction du type de noyau (paramètre α_t), de la discrétisation (paramètres α_h et N) et des paramètres cinématiques $U_{\max}T_{\text{réel}}\ /\ H_{\max}$ et (quoique plus faiblement) Fr. Des validations de (156) seront présentées en 3e partie.

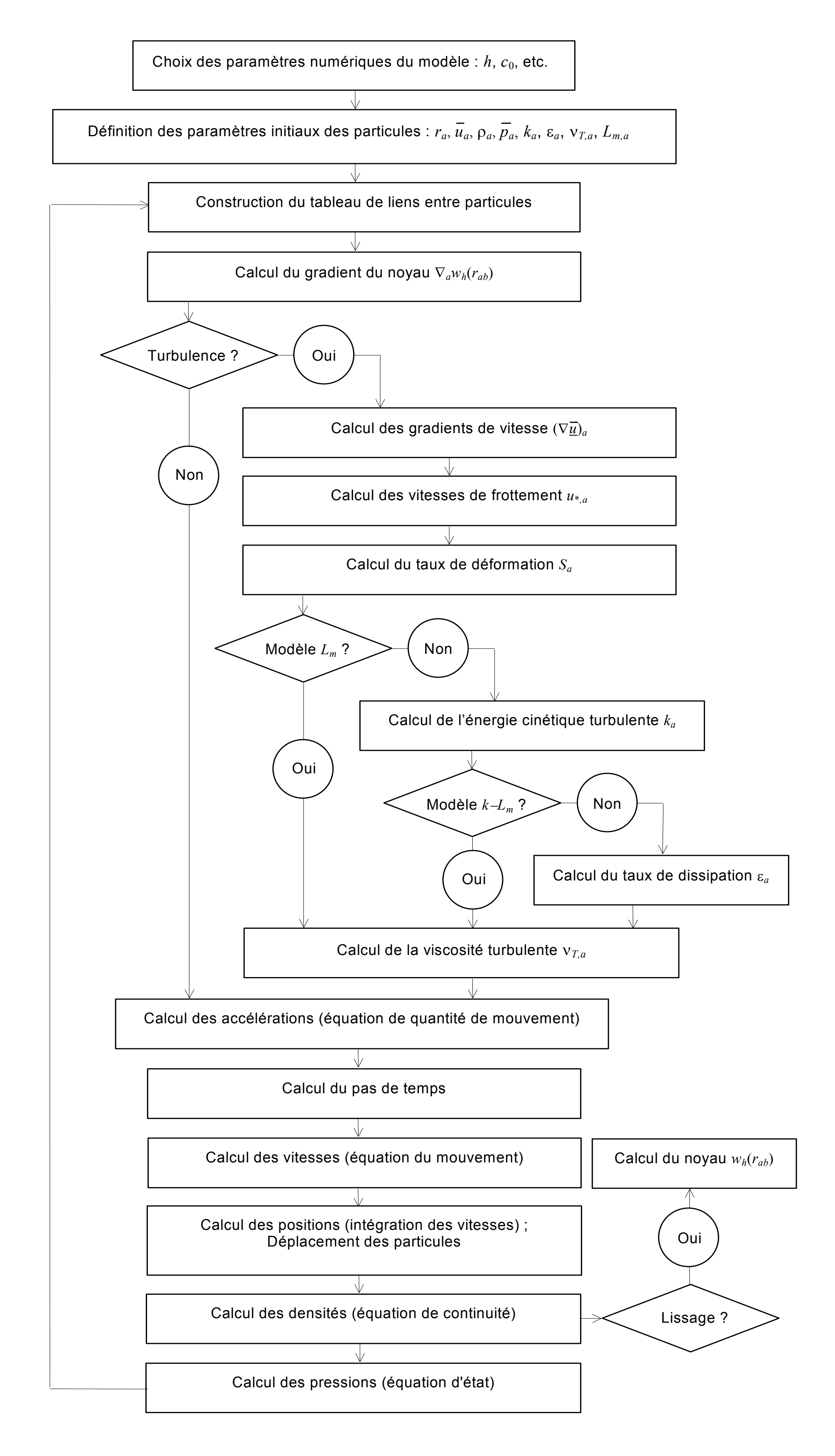


Figure 13 – Algorithme de SPARTACUS-2D V1P2.

ANNEXE 1.1 : NOTATIONS ET SYMBOLES

Ce tableau résume les principales notations utilisées dans ce document. On prendra garde au fait que l'on travaille ici en deux dimensions, si bien que les unités relatives à la notion de volume sont affectées (Cf. remarque en bas de la page 11). Ainsi, les unités de densité et de pression sont modifiées.

Notation	Signification	Unité
a, b, c	Indices particulaires	-
A, B, C	Champs scalaires	?
$\underline{A}, \underline{B}$	Champs vectoriels	?
$\underline{\underline{A}}, \underline{\underline{B}}$	Champs tensoriels d'ordre 2	?
A_a	Valeur de la quantité scalaire A au point occupé par la particule a	?
$\underline{A}_a$	Valeur de la quantité vectorielle A au point occupé par la particule a	?
$\underline{\underline{A}}_a$	Valeur de la quantité tensorielle A au point occupé par la particule a	?
A_{ab}	$A_a - A_b$	?
$\underline{A}_{ab}$	$\underline{A}_a - \underline{A}_b$	?
C_μ	Constante du modèle k–ε (= 0,09)	-
$C_{\varepsilon 1}$	Constante du modèle k–ε (= 1,44)	-
$C_{\varepsilon 2}$	Constante du modèle k–ε (= 1,92)	-
C_*	Constante de la loi de paroi rugueuse (= 8,5)	-
c_0	Vitesse du son à la densité de référence	ms^{-1}
d / dt	Dérivée lagrangienne	s^{-1}
E	Energie totale d'un système	$kg.m^2s^{-2}$
e_a	Energie interne spécifique de la particule a	m^2s^{-2}
$\underline{F}$	Fonction de dissipation d'un système mécanique	m^2s^{-3}
$\underline{F}_a$	Force reçue par la particule a	$kg.ms^{-2}$
$\underline{F}_{b\to a}$	Force exercée par b sur a	$kg.ms^{-2}$
$\underline{F}_a^p$	Force de paroi par unité de masse reçue par la particule a	ms^{-2}
$\underline{F}_{b\to a}^p$	Force de paroi par unité de masse exercée par b sur a	ms^{-2}
$\underline{F}_a^e$	Forces extérieures par unité de masse reçues par la particule a	ms^{-2}
F_a	Norme des forces totales reçues par a	$kg.ms^{-2}$

f	Fonction définissant un noyau sphérique	-
g	Accélération de la gravité	ms^{-2}
H	Hauteur d'eau	m
$H_{\max}$	Hauteur d'eau maximale	m
h	Longueur d'atténuation	m
h_t	Largeur du support du noyau	m
$\underline{\underline{I}}$	Tenseur identité	-
k	Champ d'énergie cinétique turbulente	m^2s^{-2}
k_a	Energie cinétique turbulente de la particule a	m^2s^{-2}
k_{ab}	$k_a - k_b$	m^2s^{-2}
k_s	Epaisseur de rugosité des parois	m
L	Lagrangien d'un système mécanique	$kg.m^2s^{-2}$
L_m	Longueur de mélange	m
$L_{m,a}$	Longueur de mélange de la particule a	m
M_0	Nombre de Mach de référence	-
m_a	Masse de la particule a	kg
$\underline{n}_a$	Vecteur unitaire normal à une paroi à la particule de bord a	-
P	Production d'énergie cinétique turbulente	m^2s^{-3}
P_a	Production d'énergie cinétique turbulente de la particule a	m^2s^{-3}
p	Champ de pression instantanée	$kg.s^{-2}$
$\overline{p}$	Champ de pression moyenne	$kg.s^{-2}$
p_a	Pression instantanée de la particule a	$kg.s^{-2}$
$\overline{p}_a$	Pression moyenne de la particule a	$kg.s^{-2}$
q	Rapport r / h	-
$\underline{q}_{p,ba}$	Flux de pression de b vers a	$kg^{-1}ms^{-2}$
$\underline{q}_{v,ba}$	Flux visqueux de b vers a	$kg^{-1}ms^{-2}$
$\overline{\underline{q}}_{p,ba}$	Flux de pression moyenne de b vers a	$kg^{-1}ms^{-2}$
$\overline{\underline{q}}_{v,ba}$	Flux visqueux moyen de b vers a	$kg^{-1}ms^{-2}$
$\underline{\underline{R}}$	Tenseur des contraintes de Reynolds	m^2s^{-2}
$\underline{r}$	Vecteur position d'un point quelconque de l'espace	m

$\underline{r}_a$	Vecteur position de la particule *a*	m
$\underline{r}_{ab}$	$\underline{r}_a - \underline{r}_b$	m
r_{ab}	Norme de $\underline{r}_{ab}$	m
S	Taux de déformation	s^{-1}
S_a	Taux de déformation de la particule *a*	s^{-1}
$\underline{\underline{S}}$	Tenseur taux de déformation	s^{-1}
$\underline{\underline{S}}_a$	Tenseur taux de déformation de la particule *a*	s^{-1}
$\underline{t}_a$	Vecteur unitaire tangent à une paroi à la particule de bord *a*	-
U	Energie potentielle	$kg.m^2s^{-2}$
$\underline{u}$	Champ de vitesse instantanée	ms^{-1}
$\overline{\underline{u}}$	Champ de vitesse moyenne	ms^{-1}
$\underline{u}'$	Champ de vitesse fluctuante	ms^{-1}
$\underline{u}_a$	Vitesse instantanée de la particule *a*	ms^{-1}
$\underline{u}_{ab}$	$\underline{u}_a - \underline{u}_b$	ms^{-1}
$\overline{u}_{est,a}$	Vitesse estimée pour la loi de paroi	ms^{-1}
$\overline{\underline{u}}_a$	Vitesse moyenne de la particule *a*	ms^{-1}
$\overline{\underline{u}}_{ab}$	$\overline{\underline{u}}_a - \overline{\underline{u}}_b$	ms^{-1}
u_*	Vitesse de frottement	ms^{-1}
$u_{*,a}$	Vitesse de frottement au niveau de la particule de bord *a*	ms^{-1}
u_{max}	Vitesse maximale au sein d'un écoulement	ms^{-1}
w_h	Noyau	m^{-2}
α_h	Rapport h_t / h	-
α_t	Rapport $h / \delta r$	-
α_{test}	Rapport $\delta_{est} / \delta r$	-
α_{ab}, β_{ab}	Coefficients de frottement interparticulaires	$kg.s^{-1}$, $kg.m^{-1}s^{-1}$
$\underline{\underline{\alpha}}_{ab}$	Tenseur de frottement interparticulaire entre *a* et *b*	$kg.s^{-1}$
γ	Exposant de la loi d'état	-
δ	Distance entre les particules de bord et la paroi réelle	m
$\delta(\underline{r})$	Distribution de Dirac	m^{-2}
δ_{ab}	Symbole de Kronecker	-

δ_{est}	Distance d'estimation de la vitesse pour la loi de paroi	m
$\delta(\underline{r})$	Distribution de Dirac	m^{-2}
δr	Distance initiale entre deux particules voisines	m
δt	Pas de temps	s
ε	Champ de taux de dissipation d'énergie	m^2s^{-3}
ε_a	Taux de dissipation d'énergie de la particule *a*	m^2s^{-3}
ε_{ab}	$\varepsilon_a - \varepsilon_b$	m^2s^{-3}
ε_ρ	Facteur de lissage de la densité	-
κ	Constante de Karman (= 0,41)	-
ν	Viscosité cinématique	m^2s^{-1}
ν_T	Viscosité turbulente (cinématique)	m^2s^{-1}
$\nu_{T,a}$	Viscosité turbulente de la particule *a*	m^2s^{-1}
ρ_a	Densité de la particule *a*	$kg.m^{-2}$
ρ_0	Densité de référence	$kg.m^{-2}$
σ_k	Constante du modèle k–ε (= 1,0)	-
σ_ε	Constante du modèle k–ε (= 1,3)	-
τ_a	« Volume » de la particule *a*	m^2
∇	Opérateur gradient	m^{-1}
$\nabla\cdot$	Opérateur divergence	m^{-1}
$\nabla_a w_h(r_{ab})$	Gradient du noyau	m^{-3}
.	Produit scalaire : $\underline{A}\cdot\underline{B} = A_iB_i$	-
:	Double produit contracté de deux tenseurs : $\underline{\underline{A}} : \underline{\underline{B}} = A_{ii}B_{ii}$	-
$\otimes$	Produit tensoriel de deux vecteurs : $\underline{A} \otimes \underline{B} = A_iB_j\ \underline{e}_i \otimes \underline{e}_j$	-

ANNEXE 1.2 : REFERENCES DE LA PREMIERE PARTIE

[1] Guimet V. et Laurence D. (2002), *A linearised turbulent production in the k–ε model for engineering applications*, Proceedings du 5e International Symposium on Engineering Turbulence Modelling and Measurements, Majorque, Espagne.

[2] Halem M. et Issa R. (2006), Développements d'algorithmes de lissage spatio-temporel au sein du logiciel Spartacus-2D, rapport EDF R&D / LNHE n° HP-75/06/010/A.

[3] Issa R. (2005), *Numerical assessment of the Smoothed Particle Hydrodynamics gridless method for incompressible flows and its extension to turbulent flows*, rapport de Thèse de l'Université de Manchester, Department of Mechanical, Aerospace and Manufacturing Engineering.

[4] Issa R, Lee E-S, Violeau D, Laurence D. (2004), *Incompressible separated flows simulations with the Smoothed Particle Hydrodynamics gridless method*, Int. Journ. Num. Meth. Fluids **47**:1101–1106.

[5] Launder B.E., Reece G.J. et Rodi W. (1975), *Progress in the development of Reynolds stress turbulent closure*, Journ. Fluid Mech. **68**:537.

[6] Lee E.S. (2006), *Development and applications of a truly incompressible method in Smoothed Particle Hydrodynamics*, rapport de 1e année de Thèse de l'Université de Manchester, Department of Mechanical, Aerospace and Manufacturing Engineering.

[7] Monaghan J.J. (1992), *Smoothed Particle Hydrodynamics*, Annual Review of Astrophysics **30**:543-74.

[8] Monaghan J.J. (1994), *Simulating free surface flows with SPH*, Journ. Comput. Physics, **110**:399-406.

[9] Monaghan J.J. (1995), *Simulating gravity currents with SPH*, reports and preprints from the Monash University (Australie).

[10] Morris J.P., Fox P.J. et Zhu Y. (1997), *Modelling low Reynolds number incompressible flows using SPH*, Journal of Computational Physics, **136**:214-226.

[11] Shao S. et Gotoh H. (2004), *Simulating coupled motion of progressive wave and floatting curtain wall by SPH–LES model*, Coastal Engineering Journal **46**:171–202.

[12] Viollet P.-L., Chabard J.-P, Esposito P. et Laurence D. (1998), *Mécanique des fluides appliquée. Ecoulements incompressibles dans les circuits, canaux, rivières, autour de structures et dans l'environnement*, Presses de l'Ecole Nationale des ponts et Chaussées.

[13] Violeau D. (2000), *Notions fondamentales de Mécanique lagrangienne. Application à la méthode numérique particulaire SPH,* rapport EDF R&D / LNHE n° HP-72/2000/072/A.

[14] Violeau D. (2004), *Cours d'Hydraulique à Surface Libre du LNHE. Les fondements de la turbulence, phénoménologie et modélisation dans les écoulements de fluides incompressibles*, présentation Powerpoint.

[15] Violeau D. (2004), *One and two-equations turbulent closures for Smoothed Particle*

Hydrodynamics, Proceedings de la VI[e] International Conference on Hydroinformatics, Singapour, 87-94.

[16] Violeau D. et Issa R. (2007), *Numerical modelling of complex turbulent free surface flows with the SPH gridless method: an overview*, Int. J. Num. Meth. Fluids **53**(2):277-304.

2. DEUXIEME PARTIE : DESCRIPTIF INFORMATIQUE ET MODE D’EMPLOI

2.1 DESCRIPTIF INFORMATIQUE

2.1.1 Généralités

2.1.1.1 Environnement informatique

Le code SPARTACUS-2D version 1.2 est rédigé en FORTRAN77. Dans ce chapitre, les opérations menées par le code sont successivement examinées, et les différentes routines y sont introduites, ainsi que les fichiers auxiliaires et les principales variables (dont les unités éventuelles sont toujours celles du système international, rappelées au **§ 2.3.3** avec les caractéristiques de toutes les variables). On trouvera au **§ 2.2.2** un résumé des tâches de ces routines ainsi qu'un arbre des appels. Le **§ 2.3** est consacré à la liste des mots-clefs du code.

Remarque : les particules sont indexées par des variables entières notées I ou J, et non pas *a* ou *b* comme dans l'écriture théorique des équations. Nous retiendrons cette nomenclature dans cette partie.

2.1.1.2 Commentaires sur l'écriture des routines FORTRAN

Chaque routine est munie de commentaires appropriés bilingues (Français et Anglais), ainsi que d'un en-tête se présentant comme suit :

```
C-----------------------------------------------------------------------
C                         MAIN VARIABLES
C .________________.____.______________________________________________
C !                !    !                                              !
C !   NAME         ! MODE !                   MEANING                  !
C !________________!______!____________________________________________!
C !                !    !                                              !
C ! KPARMOB        ! -->  ! MOVING WALL OR EDGE PARTICLE TYPE          !
C ! NPART          ! -->  ! TOTAL PARTICLE NUMBER                      !
C ! NPMAX          ! -->  ! MAXIMUM PARTICLE NUMBER                    !
C ! TEMPS          ! -->  ! PHYSICAL TIME                              !
C ! VXMOB,         ! <--  ! WALL VELOCITY COMPONENTS                   !
C ! VZMOB                                                              !
C !________________!______!____________________________________________!
C
C MODE : -->(NON MODIFIED DATA), <--(RESULT), <-->(MODIFIED DATA)
C-----------------------------------------------------------------------
C
C SPARTACUS2D V1P0
C D. Violeau           & R. Issa
C +33(0)1-30-87-78-31 // +33(0)1-30-87-84-28
C LNHE - 2005
C
C FONCTION : impose une vitesse a une ou plusieurs parois
C FUNCTION : computes the velocity of one or several walls
C
C PROGRAMMES APPELANT : MOUVEMENT
C CALLED BY
C
C PROGRAMMES APPELES  : -
C CALLED PROGRAMS
C
C-----------------------------------------------------------------------
```

Cette nomenclature est conforme à celle retenue pour le système hydro-informatique TELEMAC d'EDF R&D / LNHE. On y trouve une liste des variables classées par ordre alphabétique, suivies chacune d'une flèche indiquant s'il s'agit d'une variable d'entrée non modifiée (-->), d'un résultat (<--) ou d'une donnée modifiée (<-->), puis de sa description (identique à celle du **§ 2.3.3**). Dans la routine `spartacus2d.f`, jouant le rôle de programme principal, le tableau concerne l'ensemble des variables du code, tandis que les flèches de la deuxième colonne sont remplacées par leurs unités. Enfin, l'en-tête de chaque routine présente quelques précisions concernant le codage, notamment le rôle de la routine, le nom de la routine appelante et des routines appelées (s'il y a lieu).

Dans les appels, les variables sont placées dans l'ordre suivant : entiers (commençant toujours par les lettres I ou N), variables de choix (entiers ou logiques, commençant toujours par la lettre K), réels en double précision puis noms de fichiers (commençant toujours par le suffixe FICH). On en trouvera des exemples au **§ 2.3.3**. Dans chacune de ces catégories, les variables sont ensuite classées par ordre alphabétique.

Enfin, la programmation est réalisée de manière aussi claire que possible, les étapes identifiées, les variables d'appel alignées, etc., afin de permettre un usage rapide et un développement facilité.

2.1.2 Fichiers auxiliaires

Pour réaliser une simulation, l'utilisateur a besoin de différents fichiers auxiliaires, décrit dans ce paragraphe. Ce sont : le fichier des paramètres, le fichier FORTRAN, le fichier d'initialisation et éventuellement un fichier maillage.

2.1.2.1 Fichier des paramètres

Le fichier des paramètres contient toutes les information relatives aux choix des options de calcul (physiques, numériques, etc.), et doit être rédigé sous forme de fichier texte par l'utilisateur selon les principes du système TELEMAC, rappelés ci-après.

Le fichier des paramètres représente en quelque sorte le tableau de bord du calcul. Il contient un ensemble de mots clés (parmi ceux de la liste du **§ 2.3**) auxquels sont affectées des valeurs. Par exemple, la commande `PAS DE TEMPS = 10.` permet de spécifier que la valeur du pas de temps du calcul δt est de 10 secondes. Si un mot clé ne figure pas dans ce fichier, SPARTACUS-2D lui affectera la valeur par défaut définie dans le fichier dictionnaire (voir description au **§ 2.3.1**) ou dans la liste du **§ 2.3**.

SPARTACUS-2D lit le fichier des paramètres en début de calcul. Sa lecture, ainsi que celle du fichier dictionnaire, se font par un utilitaire appelé DAMOCLES, inclus dans Spartacus-2D . De ce fait, il est nécessaire, lors de la création du fichier des paramètres, de respecter les règles syntaxiques de DAMOCLES. Ces règles sont décrites ci-dessous ; un exemple fait l'objet de l'**Annexe 2.1**.

- Les mots clés peuvent être de type Entier, Réel, Logique ou Caractère.
- L'ordre des mots clés dans le fichier des paramètres est sans importance.
- Chaque ligne est limitée à 72 caractères. Cependant on peut passer à la ligne autant de fois que l'on veut à condition que le nom du mot clé ne soit pas à cheval sur deux lignes.

- Pour les mots clés de type tableau, le séparateur des valeurs successives est le point-virgule. Par exemple :

```
DENSITES DE REFERENCE = 1000.;1200.
```

- Les symboles ":" ou "=" s'emploient indifféremment comme séparateur du nom d'un mot clé et de sa valeur. Ils peuvent être précédés ou suivis d'un nombre quelconque de blancs. La valeur elle-même peut figurer à la ligne suivante. Par exemple :

```
PAS DE TEMPS = 10.
```

ou

```
PAS DE TEMPS : 10.
```

ou encore

```
PAS DE TEMPS =
10.
```

- Les caractères situés entre deux "/" sur une ligne sont considérés comme des commentaires. De même, les caractères situés entre un "/" et une fin de ligne sont considérés comme commentaires. Par exemple :

```
MODELE DE TURBULENCE = 3     / Modèle k-epsilon
```

- Une ligne commençant par un "/" en première colonne est considérée tout entière comme un commentaire, même s'il y a un autre "/" dans la ligne. Par exemple :

```
/ Le fichier FORTRAN est ./test/test1.f
```

- Ecriture des entiers : ne pas dépasser la taille maximum admise par la machine (sur une machine à architecture 32 bits, les valeurs extrêmes sont –2147483647 à +2147483648. Ne pas mettre de blanc entre le signe (facultatif pour le +) et le nombre. Un point à la fin du nombre est toléré.
- Ecriture des réels : Point et virgule sont acceptés comme séparateur décimal, ainsi que les formats E et D du Fortran. Par exemple, `1.E-3`, `0.001`, `0,001` et `1.D-3` représentent la même valeur.
- Ecriture des valeurs logiques : Sont acceptées les valeurs `1`, `OUI`, `YES`, `.TRUE.`, `TRUE`, `VRAI` d'une part, et `0`, `NON`, `NO`, `.FALSE.`, `FALSE`, `FAUX` d'autre part.
- Ecriture des chaînes de caractères : Les chaînes comportant des blancs ou des symboles réservés ("/", ":", "=", "&") doivent être placées entre cotes ('). La valeur d'un mot clé caractère peut contenir jusqu'à 144 caractères. Comme en Fortran, les cotes contenues dans une chaîne doivent être doublées. Une chaîne ne peut pas commencer ou finir par un blanc. Par exemple :

```
FICHIER FORTRAN = './test/test1.fr'
```

En plus des mots clés, un certain nombre de directives ou méta commandes interprétées au cours de la lecture séquentielle du fichier des paramètres peuvent aussi être utilisées :

- La commande `&FIN` indique la fin du fichier (même si le fichier n'est pas terminé). Cela permet de désactiver certains mots clés simplement en les plaçant derrière cette commande afin de pouvoir les réactiver facilement par la suite. Cependant, le calcul continue.
- La commande `&ETA` imprime la liste des mots clés et la valeur qui leur est affectée au moment où DAMOCLES rencontre cette commande. Cet affichage aura lieu en début de listing de sortie (voir § **2.4.2.2**).
- La commande `&LIS` imprime la liste des mots clés. Cet affichage aura lieu en début de listing de sortie (voir § **2.4.2.2**)
- La commande `&IND` imprime la liste détaillée des mots clés. Cet affichage aura lieu en début de listing de sortie (voir § **2.4.2.2**).
- La commande `&STO` provoque l'arrêt du programme, le calcul ne continuant pas.

Le nom du fichier des paramètres est donné lui-même à l'aide du mot clé : `FICHIER DES PARAMETRES` (Cf. § **2.3**). Il est lu par la routine lecdon_spartacus2d.f.

2.1.2.2 Fichier FORTRAN

Si l'utilisateur souhaite modifier des routines pour son usage personnel (pour le déroulement d'un cas particulier), il doit le faire dans le fichier FORTRAN. Dans ce cas, ce fichier doit inclure les routines que l'utilisateur modifie, et doit être déclaré dans le fichier des paramètres sous le mot-clef `FICHIER FORTRAN` (Cf. § **2.3**).

Parmi les routines du code SPARTACUS-2D, seules certaines sont ainsi accessibles à l'utilisateur, disponibles dans le répertoire /sources_util/ du code. Elles sont indiquées au § **2.2.2.1**. Lors du lancement du code, les sources modifiées seront compilées par le lanceur avec les sources non modifiées présentes dans le répertoire /sources/.

Si l'utilisateur ne souhaite modifier aucune routine, le fichier FORTRAN ne peut cependant être vide ; il doit *a minima* contenir une routine, par exemple la routine principale spartacus2d.f qui permet de définir les nombres maximaux de particules considérées (voir § **2.4.2.1**).

2.1.2.3 Fichier d'initialisation

Le fichier d'initialisation contient toutes les données relatives aux particules à l'instant initial du calcul. Il doit être rédigé par l'utilisateur au format numérique ASCII, et peut être généré par la routine geometrie.f disponible dans /sources_util/ et modifiable par l'utilisateur (Cf. § **2.4.2.1**). Il contient un en-tête spécifiant successivement différentes grandeurs. Leurs noms de variables (en majuscules dans le texte qui suit, ainsi que dans toute la suite du document) sont explicitées en détails au § **2.3.3**, et leurs formats respectifs doivent être respectés. Enfin, des détails supplémentaires sont indiqués au § **2.4.2.1** pour l'utilisateur. Ces paramètres sont :

- Le nombre de particules fluides NPARF ;
- Le nombre de particules de bord NPARB ;
- Le nombre total de particules NPART ;

- La distance initiale entre deux particules voisines DR.

Puis, sur autant de lignes qu'il y a de particules, on trouve pour chacune d'entre elles :

- Le numéro de la particule I ;
- Ses coordonnées X(I) et Z(I) ;
- Dans le cas d'une particule de bord, l'angle THETA(I–NPARF) compté *en degrés trigonométriques* que fait la normale *intérieure* au bord avec l'axe *x* au point occupé par la particule de bord I ;
- Un label KPAR(I) désignant respectivement une particule fluide (KPAR(I) = 1), une particule de bord (KPAR(I) = 2) ou une particule fictive (KPAR(I) = 3) ;
- Le nombre KFLUID(I) correspondant au type de fluide de la particule I, pour un écoulement polyphasique (ce nombre peut prendre les valeurs 1, 2, etc. jusqu'à NFLUID) ;
- Un entier KENT(I) indiquant si I est une particule fluide dont la vitesse est imposée par l'utilisateur. Dans ce cas, KENT(I) = 1.
- Dans le cas d'une particule de bord ou fictive, un entier KPARMOB(I) indiquant s'il s'agit d'une paroi mobile (KPARMOB(I) = 1) ou non (KPARMOB(I) = 0).

Remarque importante : les particules de bord (KPAR(I) =2) doivent être rangées après les particules fluides (c'est-à-dire pour I allant de NPARF + 1 à NPARF + NPARB), et doivent précéder les particules fictives (KPAR(I) = 3, pour I allant de NPARF + NPARB + 1 à NPART) qui terminent donc la liste. C'est la raison pour laquelle le tableau THETA, indexé de 1 à NPARB, est lu pour I = NPARF + 1 à NPARF + NPARB en décalant vers la gauche de l'entier NPARF.

Un exemple de fichier d'initialisation est donné en **Annexe 2.1**. Son nom est indiqué dans le fichier des paramètres par le mot-clef `FICHIER D'INITIALISATION` (Cf. § **2.3**). Il est lu par le code dans la routine lecinit.f.

Comme nous l'avons évoqué, le fichier d'initialisation peut être créé à l'aide de geometrie.f (Cf. § **2.4.2.1**). Toutefois, dans le cas d'une suite de calcul, le fichier d'initialisation est automatiquement généré par le calcul précédent ; il s'agit en effet du « fichier suite » correspondant au calcul précédent (Cf § **2.1.3.3**). Son contenu est alors différent, mais l'utilisateur n'a pas à s'en soucier, et doit seulement indiquer dans le fichier des paramètres, comme nom de fichier d'initialisation, le nom du fichier suite du calcul précédent (Cf. § **2.4.2.7**).

2.1.2.4 Fichier maillage

Le fichier maillage est un fichier binaire contenant les données d'un maillage de type « éléments finis » au format SELAFIN de Rubens©, et peut être généré avec le logiciel MATISSE du système TELEMAC (Peron et Prygoda, 1997). A ce titre, il ne sert que si ce dernier est utilisé comme post-processeur, afin d'interpoler les champs dans le fichier des résultats (Cf. §§ **2.1.3.2** et **2.4.3.1**) grâce à la routine interpol.f. Il doit être déclaré dans le fichier des paramètres sous le mot-clef `FICHIER DE MAILLAGE` (§ **2.4.2.1**).

2.1.3 Fichiers résultats

Les fichiers résultats sont au nombre de deux, l'un contenant les positions des particules, l'autre la listes des variables qu'elles portent, sous forme de champs associés.

2.1.3.1 Fichier des positions

Le fichier des positions (format ASCII) est rédigé par la routine ecrit.f et doit être déclaré dans le fichier des paramètres sous le mot-clef `FICHIER DES POSITIONS` (Cf. **§ 2.3**). Il contient d'abord quelques lignes d'en-tête sans intérêt pour le programmeur ou l'utilisateur. Puis on trouve, pour chaque instant de sortie des positions (choisi par l'utilisateur), un bloc contenant le temps correspondant TEMPS puis, pour chaque particule I dans l'ordre de leur numérotation :

- Le numéro de la particule I (pour le format Rubens© seulement) ;
- Son abscisse X(I) ;
- Son ordonnée Z(I).

Ce fichier peut être exploité par l'utilisateur grâce aux logiciels de post-traitement Rubens© et Tecplot©. On trouvera les détails permettant son utilisation au **§ 2.4.3**.

2.1.3.2 Fichier des résultats

Le fichier des résultats proprement dit doit être déclaré dans le fichier des paramètres sous le mot-clef `FICHIER DES RESULTATS` (Cf. **§ 2.3**). Il contient, pour chaque instant de sortie des champs (choisi par l'utilisateur), la liste de toutes les variables que l'utilisateur peut visualiser sous forme de champs spatiaux (pression des particules, vitesses, viscosité turbulente, etc.), dont la liste est donnée au **§ 2.4.3**. Ce fichier est rédigé par la routine interpol.f au format binaire pour le post-processeur Rubens© (les champs y sont en effet interpolés sur le fichier maillage décrit au **§ 2.1.2.4**) , et par la routine ecrit.f au format ASCII pour Tecplot©.

2.1.3.3 Fichier suite

Le fichier suite (fichier texte) est lui aussi rédigé par la routine ecrit.f et doit être déclaré dans le fichier des paramètres sous le mot-clef `FICHIER SUITE` (Cf. **§ 2.3**). Contenant les données du système à l'issue d'un calcul, il jouera le rôle de fichier d'initialisation lors d'un nouveau calcul se présentant comme la suite du précédent (Cf. **§ 2.4.2.7**). Il contient un en-tête au même format que celui du fichier d'initialisation classique décrit au **§ 2.1.2.3**, puis, sur autant de lignes qu'il y a de particules, on trouve pour chacune d'entre elles les données suivantes, correspondant au dernier pas de temps :

- La masse de la particule I ;
- Ses coordonnées X(I) et Z(I) ;
- Les composantes de sa vitesse VX(I) et VZ(I) ;
- Sa densité RHO(I) ;
- Sa pression P(I) ;
- Son énergie cinétique turbulente TKE(I) ;
- Son taux de dissipation d'énergie EPS(I) ;

- Sa viscosité turbulente NUT(I) ;
- Les quantités THETA(I–NPARF), KPAR(I), KFLUID(I) , KENT(I) et KPARMOB(I) déjà définies au **§ 2.1.2.3**.

2.2 DEROULEMENT D'UN CALCUL

Après les initialisations décrites plus haut, la boucle en temps peut commencer. Pour chaque itération IT allant de 1 à NT (nombre de pas de temps demandé par l'utilisateur), les opérations décrites ci-après sont effectuées.

2.2.1 Liste des opérations effectuées

La routine principale est spartacus2d.f. On trouvera dans ce paragraphe la liste des opérations effectuées par le code, résumées en **figure 14** sous forme d'un arbre des appels.

2.2.1.1 Initialisations

Les paramètres du calcul, regroupés dans le fichier des paramètres (voir **§ 2.1.2.1**) sont lus par la routine lecparam.f, puis les paramètres géométriques (positions initiales des particules, angles des normales aux bords, etc.) sont acquises dans le fichier d'initialisation (**§ 2.1.2.3**) par la routine lecinit.f, appelée par initial.f (le cas d'une suite de calcul est examiné au **§ 2.4.2.7**). Les quantités physiques y sont également initialisées de la manière suivante : les densités sont initialisées en fonction du type de fluide de chaque particule et des densités de référence de chaque fluide. Les masses sont calculées à partir des densités, en considérant que chaque particule possède un « volume » (en fait une surface, en dimension 2) égal au carré de δr, écart des particules voisines dans leur disposition initiale. La pression initiale n'a pas besoin d'être définie (elle est nulle), car elle ne fait pas l'objet d'un schéma numérique, étant calculée à partir de l'équation d'état (129). En revanche, l'utilisateur a la possibilité de définir dans initial.f les vitesses initiales des particules s'il le désire (elles sont nulles par défaut). Initial.f appelle aussi la routine melange.f, modifiable par l'utilisateur, qui définit le tableau LM des longueurs de mélange. Par défaut, elle est choisie constante égale à δr, ce qui consiste en une sorte de « LES bidimensionnelle » (voir le **§ 1.1.3.2**). Enfin, c'est toujours dans initial.f que le pas de temps est initialisé d'après la seconde condition de (149).

La routine ouverture.f se charge ensuite de l'ouverture des fichiers résultats et de rédiger leurs en-têtes respectifs (**§2.1.3**). Enfin la routine privee.f permet le calcul d'une variable de sortie privée (tableau PRIVE) définie par l'utilisateur s'il le souhaite (voir **§ 2.4.3**).

La routine tablien.f permet de déterminer le tableau de liens entre particules, grâce à l'algorithme décrit au **§ 2.2.2**. Cette routine est appelée en tout début de calcul si Rubens© est choisi comme post-processeur, afin de procéder à une interpolation initiale des champs sur le maillage dans la routine interpol.f. Enfin, la routine ecrit.f réalise une écriture initiale dans les fichiers résultats.

2.2.1.2 Tableau de liens et gradient du noyau

La boucle en temps peut alors commencer. La routine tablien.f, introduite au paragraphe précédent, est d'abord appelée. Deux tableaux sont fournis en sortie, donnant, pour chaque particule fluide

d'indice I, le nombre de particules d'indices *supérieurs* à I liées à I (tableau NLIEN), et les indices de ces particules (tableau ILIEN). Cette restriction est conforme au procédé d'exploitation des symétries explicité au **§1.2.2.2**. Les particules de bord et fictives ont donc moins de liens, puisque leurs indices sont placés en fin de liste ; elles sont en revanche elles-mêmes liens de certaines particules fluides. En plus des tableaux de liens NLIEN et ILIEN, on estime alors les tableaux XAB, ZAR, RAB, VXAB et VZAB (Cf. § **2.3.3**). Un traitement spécial est réservé au cas d'un écoulement périodique selon x (§ **1.2.2.1**).

Les valeurs du gradient du noyau sont évaluées dans kernel.f. Selon l'ordre du noyau, cette routine fait appel à kernel3.f (ordre 3, équation (13)), kernel4.f (ordre 4, équation (14)) ou kernel5.f (ordre 5, équation (15)). Ce calcul est fait uniquement pour les paires de particules I et J liées I < J).

2.2.1.3 Calcul des quantités cinématiques

Les contenus de ce paragraphe et du suivant sont conformes à l'algorithme général présenté au **§ 1.2.1.1** (équation (145)).

Si l'utilisateur fait le choix d'un modèle de turbulence, le calcul des grandeurs cinématiques commence par l'estimation du tenseur gradient des vitesses de chaque particule (équation (141), routine gradvit.f) puis du tableau des vitesses de frottement USTAR attachées aux particules de bord (Cf. **§ 1.1.4.4**, routine frottement.f).

Le cœur du traitement des grandeurs cinématique se fait dans la routine impulsion.f. Si le modèle retenu pour le traitement des bords solides est celui des forces répulsives (**§ 1.1.4.2**), on fait appel à la routine forcparois.f (équations (106) à (108)). Les forces extérieures sont ensuite estimées dans forcext.f ; on y trouve la gravité et les forces répulsives de parois (s'il y a lieu), ainsi que l'éventuelle force motrice pilotant l'écoulement en guise de moteur (cas d'un canal incliné, par exemple, comme indiqué au **§ 2.4.2.3**).

Les grandeurs turbulentes sont ensuite estimées dans viscturb.f (le **§ 2.2.1.4** est réservé à sa description détaillée). La routine impulsion.f poursuit alors en établissant les flux de pression (équations (130) ou (131), routine fluxpres.f) et de viscosité (équations (132) ou (133), routine fluxvisq.f), puis termine par l'établissement de l'équation de quantité de mouvement des particules (second membre de l'équation (127)), aboutissant à la donnée des accélérations de chaque particule (tableaux AX et AZ).

Le pas de temps (équation (149)) est calculé dans pastemps.f, puis la routine mouvement.f intègre les vitesses (tableaux VX et VZ, équation (127)). Les positions X(I) et Z(I) de chaque particule I sont ensuite déduites grâce à l'équation (128). mouvement.f fait également appel à la routine paroimobile.f s'il y a lieu, où l'utilisateur peut imposer un mouvement aux éventuelles parois mobiles (Cf. **§ 2.4.2.2**).

La densité RHO(I) de chaque particule I est calculée dans continuite.f, en intégrant l'équation (126) puis éventuellement en prenant en compte le lissage (148). Enfin les pressions P(I) sont calculées dans etat.f grâce à l'équation (129). L'utilisateur peut modifier cette routine s'il veut changer de loi d'état.

2.2.1.4 Calcul des grandeurs turbulentes

La routine viscturb.f, appelée par impulsion.f, se charge de calculer la viscosité turbulente NUT(I) de

chaque particule (équations (134) ou (135)). On commence par y estimer le tableau des taux de déformation S (équations (141) à (144)), puis on appelle la routine kequation.f si le modèle de turbulence nécessite la résolution de l'équation sur l'énergie cinétique turbulente K (équation (136)). Le taux de dissipation EPS est ensuite calculé grâce à l'équation (137) ou (138) (routine epsilon.f). Dans le cas d'un modèle de longueur de mélange ou k–L_m, le tableau des longueurs de mélange LM est lu dans melange.f.

2.2.1.5 Particules entrantes et sortantes – Opérations finales

La vitesse des particules libres, comme on l'a vu, est imposée dans mouvement.f. Cependant, si l'utilisateur souhaite imposer un flux entrant de particule à travers une frontière ouverte, il peut imposer leurs positions et vitesses respectives dans la routine entpar.f (Cf. § **2.4.2.3**). La routine sorpar.f gère la sortie des particules dont les coordonnées excèdent les valeurs maximales imposées dans le fichier des paramètres (Cf. § **2.4.2.4**).

A la fin de chaque itération, si le nombre de pas de temps écoulés est un multiple de la période pour les sorties des positions (resp. des champs), une écriture est faite dans le fichier des positions (resp. des résultats) grâce aux routines privee.f et ecrit.f. Ces périodes de sortie doivent être définies par l'utilisateur dans le fichier des paramètres (**§ 2.1.2.1**). Une écriture a toujours lieu à la fin du calcul, même si le nombre d'itérations n'est pas un multiple de l'une de ces deux périodes ; ecrit.f. procède alors à l'écriture du fichier suite (**§ 2.1.3.3**). La boucle en temps se termine ici.

Enfin, spartacus2d.f procède à la fermeture des fichiers et le calcul se termine.

2.2.2 Liste des routines et arbre des appels

Les routines de SPARTACUS-2D sont décrites ci-après dans l'ordre alphabétique avec leur description succincte, puis l'architecture des appels fait l'objet d'un arbre algorithmique conditionnel, reprenant la structure de l'algorithme théorique décrit au **§ 1.2.3.1**. Les routines accessibles à un utilisateur externe et donc modifiables pour des usages privés sont indiquées dans les deux cas.

2.2.2.1 Liste des routines

continuite.f ……………… Calcule la densité de chaque particule via l'équation de continuité (126) et l'équation numérique associée (7e ligne du système (145)).
Routine non accessible.

ecrit.f ……………………… Gère les écritures dans les fichiers des positions et des résultats et le fichier suite de calcul.
Routine non accessible.

entpar.f …………………… Gère les entrées de particules en cas de flux entrant.
➢ *Routine **accessible et modifiable** par l'utilisateur s'il souhaite imposer un flux entrant de particules à vitesses prescrites.*

epsilon.f ………………… Calcule le taux de dissipation de chaque particule via l'équation (138) et l'équation numérique associée (4e ligne du système (145) et sa condition aux parois (119)).

Routine non accessible.

etat.f .... Calcule la pression de chaque particule via l'équation d'état (129) (ou la 8e ligne du système (145)).

➢ *Routine* ***accessible et modifiable*** *par l'utilisateur s'il souhaite changer d'équation d'état.*

fluxpres.f .... Calcule les flux de pression pour chaque paire de particules liées via les équations (130) et (131).
Routine non accessible.

fluxvisq.f .... Calcule les flux visqueux pour chaque paire de particules liées via les équations (132) et (133).
Routine non accessible.

forcext.f .... Calcule les forces extérieures pour chaque particule : gravité, forces de parois, force motrice (s'il y a lieu), selon l'équation (156), p. 104.
Routine non accessible.

forcparois.f .... Calcule les forces de parois pour chaque particule, selon les équations (106) à (109).
Routine non accessible.

frottement.f .... Calcule la vitesse de frottement pour chaque particule de bord, selon les équations (121) et (122) ou (125).
Routine non accessible.

geometrie.f .... Construit le fichier d'initialisation pour un nouveau calcul. Cette routine n'est pas appelée par le code, mais figure dans /sources_util/.

➢ *Routine* ***accessible et modifiable*** *par l'utilisateur s'il souhaite créer une nouvelle disposition initiale de particules.*

gradvit.f .... Calcule les composantes du gradient de vitesse pour chaque particule, selon l'équation (141).
Routine non accessible.

impulsion.f .... Calcule l'accélération de chaque particule fluide, à partir de l'équation de quantité de mouvement (127).
Routine non accessible.

initial.f .... Initialise le calcul (constantes physiques, lectures initiales et valeurs par défaut).

➢ *Routine* ***accessible et modifiable*** *par l'utilisateur s'il souhaite modifier les paramètres initiaux des particules (notamment les vitesses).*

interpol.f .... Interpole sur une grille les grandeurs exploitées en sorties sous forme de champs continus (pour les sorties de type Rubens©).
Routine non accessible.

kequation.f .... Calcule l'énergie cinétique turbulente de chaque particule via les équations (136), (137), (139) et (140) et les équations numériques

associées (2e et 3e lignes du système (145) et condition aux parois (116)).
Routine non accessible.

kernel.f .... Calcule le gradient du noyau pour les arguments correspondant aux positions relatives des particules.
Routine non accessible.

kernel3.f .... Calcule le gradient du noyau du troisième ordre (13) pour les arguments correspondant aux positions relatives des particules.
Routine non accessible.

kernel4_val.f .... Calcule la valeur du noyau du quatrième ordre (7) pour les arguments correspondant aux positions relatives des particules.
Routine non accessible.

kernel4.f .... Calcule le gradient du noyau du quatrième ordre (14) pour les arguments correspondant aux positions relatives des particules.
Routine non accessible.

kernel5.f .... Calcule le gradient du noyau du cinquième ordre (15) pour les arguments correspondant aux positions relatives des particules.
Routine non accessible.

lecdon_spartacus2d.f ... Lit le fichier des paramètres suivant les prescriptions de DAMOCLES.
Routine non accessible.

lecinit.f .... Lit les grandeurs initiales relatives à chaque particule dans le fichier d'initialisation, dans le cas d'un premier calcul.
Routine non accessible.

lecparam.f .... Lit le fichier des paramètres en appelant lecdon_spartacus2d.f.
Routine non accessible.

lecsuit.f .... Lit les grandeurs initiales relatives à chaque particule dans le fichier d'initialisation, dans le cas d'une suite de calcul.
Routine non accessible.

melange.f .... Définit la longueur de mélange pour chaque particule, à l'instant initial.
➢ *Routine **accessible et modifiable** par l'utilisateur s'il souhaite définir une répartition particulière de la longueur de mélange.*

mouvement.f .... Résout l'équation de quantité de mouvement pour déduire les vitesses (6e ligne du système (145) et sa condition aux parois (115)), puis les positions de chaque particule, grâce à l'équation (128) et à sa forme numérique (9e et dernière ligne du système (145)).
Routine non accessible.

ouverture.f .... Gère les ouvertures de fichiers et les écritures des en-têtes.
Routine non accessible.

paroimobile.f .... Définit la vitesse des parois mobiles.
➢ *Routine **accessible et modifiable** par l'utilisateur s'il souhaite introduire*

des parois mobiles.

pastemps.f ……………… Calcule le pas de temps via l'équation (149).
Routine non accessible.

privee.f …………………… Permet de définir une variable sortie privée pour l'utilisateur.
➢ *Routine* ***accessible et modifiable*** *par l'utilisateur s'il souhaite définir une variable de sortie privée.*

sorpar.f …………………… Gère les sorties de particules à travers les bornes du domaine.
Routine non accessible.

spartacus2d.f …………… Programme principal. Il gère la boucle en temps et les appels de routines.
➢ *Routine* ***accessible et modifiable*** *par l'utilisateur.*

tablien.f …………………… Détermine les liens de chaque particule.
Routine non accessible.

tauxdef.f ………………… Calcule le taux de déformation de chaque particule à l'aide des équations (141) à (144) (ou la 1ère ligne du système (145)).
Routine non accessible.

viscturb.f ………………… Calcule la viscosité turbulente de chaque particule à l'aide des équations (134) et (135) (ou la 5e ligne du système (145)).
Routine non accessible.

2.2.2.2 Arbre des appels

L'arbre des appels visible en **figure 14** est conforme à l'algorithme général donné en **figure 13** (1e partie). Il obéit à la nomenclature suivante :

- Les routines sont indiquées par des rectangles ;
- Les rectangles grisés concernent les routines accessibles à l'utilisateur ;
- Les flèches conditionnelles (fondées sur des arguments logiques) sont pourvues de losanges ; les noms des variables logiques (Cf. **§ 2.3.3**) y sont en majuscules.

Les variables déterminant les appels conditionnels (logiques ou entiers commençant tous par un « K ») correspondent aux noms retenus dans le code (voir **§ 2.3.3**). On notera que la routine geometrie.f, disponible dans /Sources_util/ mais hors code (voir **§ 2.4.2.1**), ne figure pas sur cet arbre.

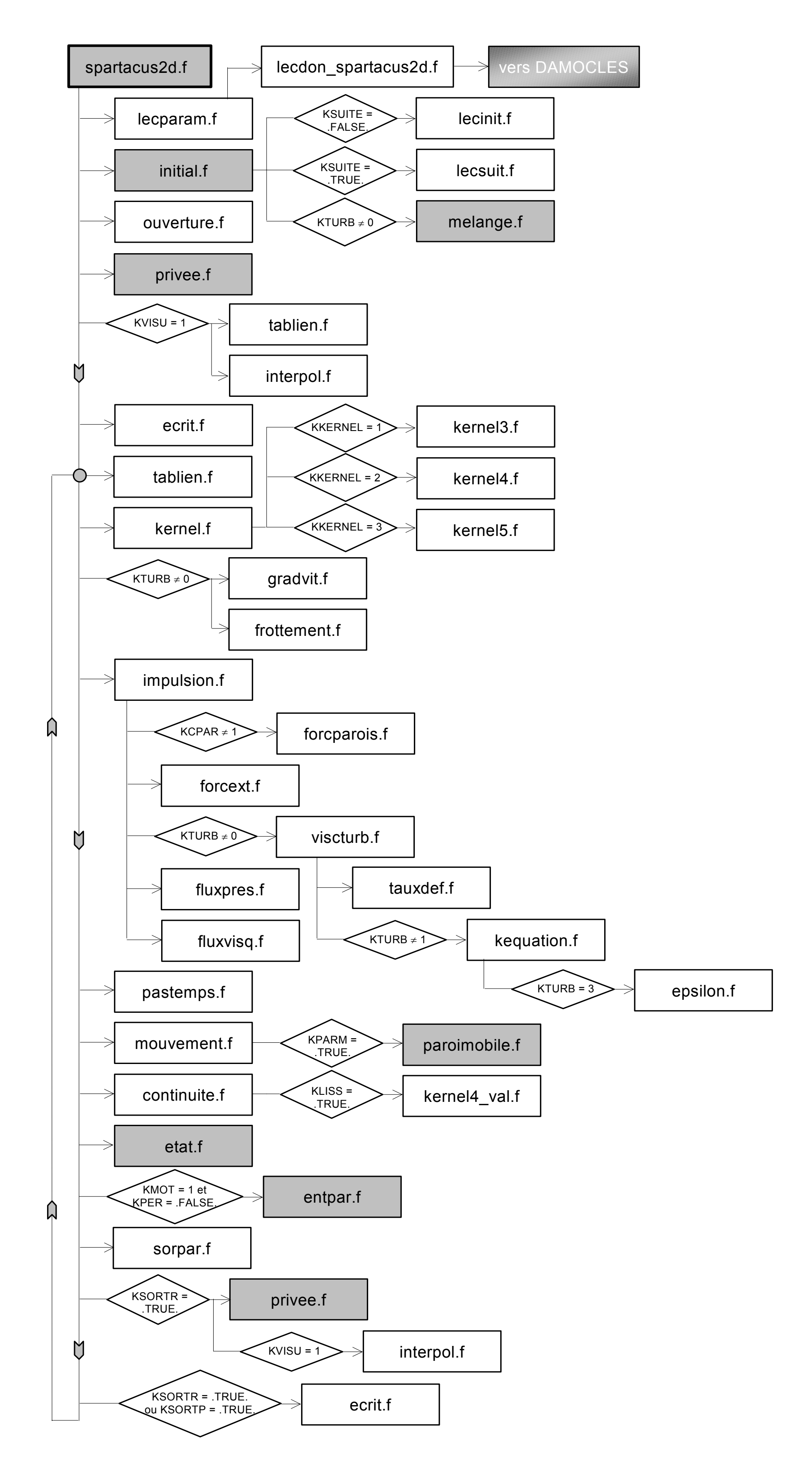


Figure 14 – Arbre des appels de SPARTACUS-2D V1P2.

2.3 DICTIONNAIRE DU CODE

2.3.1 Fichier dictionnaire

Ce fichier est accessible à l'utilisateur pour consultation. Il contient toutes les informations concernant les mots-clefs du code (classés par type), notamment celles données au **§ 2.3**, mais aussi d'autres informations utiles, comme les rubriques des mots-clefs, leurs bornes, etc. D'autre part, à la fin du fichier dictionnaire se trouvent listées des renseignements concernant les fichiers et les librairies du code.

2.3.2 Liste des mots-clefs

Les mots-clefs du logiciel SPARTACUS-2D sont listés ci-dessous dans l'ordre alphabétique, avec pour chacun une description, la variable FORTRAN qui lui est associée dans le paragraphe suivant, son type (réel, entier, liste, etc., en indiquant l'unité physique dans le cas d'un réel), sa traduction anglaise (tel qu'elle apparaît dans les cartouches des routines du code) ainsi que son mode d'utilisation (signification, valeurs possibles, valeur par défaut, etc.). On mentionne aussi les mots-clefs associés.

Certains mots-clefs sont obligatoires, repérés alors par une mention fléchée. Les autres sont conditionnés par des mots-clefs génériques qui en déterminent l'usage facultatif (par exemple, le mot-clef `MODELE DE PRODUCTION` n'a de sens que si `MODELE DE TURBULENCE` est fixé à `2` ou `3`). Pour pallier l'absence de valeurs par défaut, on indique enfin des valeurs recommandées, conformément à la première partie de ce document. On prendra garde au fait que certains mots-clefs sont obligatoires.

ABSCISSE MAXIMALE DU DOMAINE D ETUDE

➢ *Mot-clef* ***obligatoire***.

Abscisse maximale du domaine de calcul considéré.
Correspond à la variable Fortran XMAX.
Type : 1 réel (m).
Valeurs possibles : tous réels strictement supérieurs à ABSCISSE MINIMALE DU DOMAINE D'ETUDE.
Valeur recommandée : aucune.
Valeur par défaut : 10^4 m.
Version anglaise : *MAXIMAL AXIAL CO-ORDINATE OF THE DOMAIN.*
Mots-clefs associés : ABSCISSE MINIMALE DU DOMAINE D ETUDE
COTE MINIMALE DU DOMAINE D ETUDE
COTE MAXIMALE DU DOMAINE D ETUDE

ABSCISSE MINIMALE DU DOMAINE D ETUDE

➢ *Mot-clef* ***obligatoire***.

Abscisse minimale du domaine de calcul considéré.
Correspond à la variable Fortran XMIN.
Type : 1 réel (m).
Valeurs possibles : tous réels strictement inférieurs à ABSCISSE MAXIMALE DU DOMAINE D'ETUDE.

Valeur recommandée : aucune.
Valeur par défaut : -10^4 m.
Version anglaise : *MINIMAL AXIAL CO-ORDINATE OF THE DOMAIN*.
Mots-clefs associés : ABSCISSE MAXIMALE DU DOMAINE D ETUDE
COTE MINIMALE DU DOMAINE D ETUDE
COTE MAXIMALE DU DOMAINE D ETUDE

AMORTISSEMENT SELON X

- *Mot-clef* ***obligatoire***.

Logique de choix d'amortissement selon *x*.
Correspond à la variable Fortran KGAMMAX.
Type : 1 logique.
Valeurs possibles : **OUI** (ou VRAI, ou YES, ou TRUE), **NON** (ou FAUX, ou NO, ou FALSE).
Valeur recommandée : NON, sauf pour un cas statique.
Valeur par défaut : NON.
Version anglaise : *X-DAMPING*.
Mots-clefs associés : AMORTISSEMENT SELON Z
COEFFICIENT D AMORTISSEMENT SELON X
COEFFICIENT D AMORTISSEMENT SELON Z

AMORTISSEMENT SELON Z

- *Mot-clef* ***obligatoire***.

Logique de choix d'amortissement selon *z*.
Correspond à la variable Fortran KGAMMAZ.
Type : 1 logique.
Valeurs possibles : **OUI** (ou VRAI, ou YES, ou TRUE), **NON** (ou FAUX, ou NO, ou FALSE).
Valeur recommandée : NON, sauf pour un cas statique.
Valeur par défaut : NON.
Version anglaise : *Z-DAMPING*.
Mots-clefs associés : AMORTISSEMENT SELON X
COEFFICIENT D AMORTISSEMENT SELON X
COEFFICIENT D AMORTISSEMENT SELON Z

COEFFICIENT D AMORTISSEMENT SELON X

Mot-clef facultatif, à utiliser uniquement si AMORTISSEMENT SELON X vaut OUI.
Valeur du coefficient d'amortissement selon *x*.
Correspond à la variable Fortran GAMMAX.
Type : 1 réel (s^{-1}).
Valeurs possibles : tous réels positifs.
Valeurs recommandées : de l'ordre de grandeur de l'inverse du PAS DE TEMPS.
Valeur par défaut : 10^4 s^{-1}.
Version anglaise : *X-DAMPING COEFFICIENT*.
Mots-clefs associés : AMORTISSEMENT SELON X
AMORTISSEMENT SELON Z
COEFFICIENT D AMORTISSEMENT SELON Z

COEFFICIENT D AMORTISSEMENT SELON Z

Mot-clef facultatif, à utiliser uniquement si AMORTISSEMENT SELON Z vaut OUI.
Valeur du coefficient d'amortissement selon *z*.
Correspond à la variable Fortran GAMMAZ.
Type : 1 réel (s^{-1}).
Valeurs possibles : tous réels positifs.
Valeurs recommandées : de l'ordre de grandeur de l'inverse du PAS DE TEMPS.
Valeur par défaut : 10^4 s^{-1}.
Version anglaise : *Z-DAMPING COEFFICIENT*.
Mots-clefs associés : AMORTISSEMENT SELON X
AMORTISSEMENT SELON Z
COEFFICIENT D AMORTISSEMENT SELON X

COEFFICIENT DES FORCES DE PAROI

Mot-clef facultatif, à utiliser uniquement si TYPE DE CONDITIONS AUX PAROIS vaut 2
et si TYPE DE FORCE DE PAROI vaut 1.
Coefficient de la force de paroi n°1.
Correspond à la variable Fortran CPAR.
Type : 1 réel (m^2s^{-2}).
Valeurs possibles : tous réels positifs.
Valeurs recommandées : de l'ordre de grandeur de 5 à 10 fois la hauteur d'eau maximale multipliée par l'accélération de la gravité.
Valeur par défaut : 50 m^2s^{-2}.
Version anglaise : *WALL-FORCE COEFFICIENT*.
Mots-clefs associés : TYPE DE CONDITIONS AUX PAROIS
TYPE DE FORCE DE PAROI

COTE MAXIMALE DU DOMAINE D ETUDE

- *Mot-clef* ***obligatoire***.

Cote maximale du domaine de calcul considéré.
Correspond à la variable Fortran ZMAX.
Type : 1 réel (m).
Valeurs possibles : tous réels strictement supérieurs à COTE MINIMALE DU DOMAINE D'ETUDE.
Valeur recommandée : aucune.
Valeur par défaut : 10^4 m.
Version anglaise : *MAXIMAL VERTICAL CO-ORDINATE OF THE DOMAIN*.
Mots-clefs associés : ABSCISSE MINIMALE DU DOMAINE D ETUDE
ABSCISSE MAXIMALE DU DOMAINE D ETUDE
COTE MINIMALE DU DOMAINE D ETUDE

COTE MINIMALE DU DOMAINE D ETUDE

- *Mot-clef* ***obligatoire***.

Cote minimale du domaine de calcul considéré.
Correspond à la variable Fortran ZMIN.

Type : 1 réel (m).
Valeurs possibles : tous réels strictement inférieurs à COTE MAXIMALE DU DOMAINE D'ETUDE.
Valeur recommandée : aucune.
Valeur par défaut : -10^4 m.
Version anglaise : *MINIMAL VERTICAL CO-ORDINATE OF THE DOMAIN*.
Mots-clefs associés : ABSCISSE MINIMALE DU DOMAINE D ETUDE
ABSCISSE MAXIMALE DU DOMAINE D ETUDE
COTE MAXIMALE DU DOMAINE D ETUDE

DENSITES DE REFERENCE

➢ *Mot-clef* ***obligatoire***.

Tableau des densités de référence des fluides considérés.
Correspond à la variable Fortran RHO0.
Type : NFLUID réels (kg.m^{-2}) séparés par des virgules.
Valeurs possibles : tous réels strictement positifs.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *REFERENCE DENSITIES*.
Mots-clefs associés : NOMBRE DE FLUIDES
VISCOSITES MOLECULAIRES

FACTEUR DE LISSAGE

Mot-clef facultatif, à utiliser uniquement si LISSAGE DE LA DENSITE vaut OUI.

Facteur de lissage de la densité.
Correspond à la variable Fortran FACTLISS.
Type : 1 réel (-).
Valeurs possibles : tous réels positifs.
Valeurs recommandées : quelques centièmes.
Valeur par défaut : 0,05.
Version anglaise : *SMOOTHING FACTOR*.
Mot-clef associé : LISSAGE DE LA DENSITE

FICHIER INITIALISATION

➢ *Mot-clef* ***obligatoire***.

Fichier contenant la géométrie initiale du système.
Correspond à la variable Fortran FICHINIT.
Type : une chaîne de caractères.
Valeurs possibles : toute chaîne de caractères correspondant à un fichier existant.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *INITIALIZING FILE*.
Mots-clefs associés : FICHIER DE MAILLAGE
FICHIER DES PARAMETRES
FICHIER FORTRAN

FICHIER DE MAILLAGE

- *Mot-clef* ***obligatoire***.

Fichier contenant le maillage d'interpolation pour lecture sous Rubens©.
Correspond à la variable Fortran FICHMAILL.
Type : une chaîne de caractères.
Valeurs possibles : toute chaîne de caractères correspondant à un fichier existant.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *MESH FILE*.
Mots-clefs associés : FICHIER D'INITIALISATION
FICHIER DES PARAMETRES
FICHIER FORTRAN

FICHIER DES PARAMETRES

- *Mot-clef* ***obligatoire***.

Fichier contenant les paramètres du calcul.
Correspond à la variable Fortran FICHPARAM.
Type : une chaîne de caractères.
Valeurs possibles : toute chaîne de caractères correspondant à un fichier existant.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *STEERING FILE*.
Mots-clefs associés : FICHIER D'INITIALISATION
FICHIER DE MAILLAGE
FICHIER FORTRAN

FICHIER DES POSITIONS

- *Mot-clef* ***obligatoire***.

Fichier résultat contenant les positions des particules.
Correspond à la variable Fortran FICHPOS.
Type : une chaîne de caractères.
Valeurs possibles : toute chaîne de caractères correspondant à un fichier existant.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *POSITION FILE*.
Mots-clefs associés : FICHIER DES RESULTATS
FICHIER SUITE

FICHIER DES RESULTATS

- *Mot-clef* ***obligatoire***.

Fichier résultat contenant les champs continus.
Correspond à la variable Fortran FICHRES.
Type : une chaîne de caractères.
Valeurs possibles : toute chaîne de caractères correspondant à un fichier existant.
Valeur recommandée : aucune.

Valeur par défaut : aucune.
Version anglaise : *RESULT FILE*.
Mots-clefs associés : FICHIER DES POSITIONS
FICHIER DE MAILLAGE

FICHIER FORTRAN

- *Mot-clef* ***obligatoire***.

Fichier contenant les routines modifiées par l'utilisateur ou le programme principal si aucune modification n'est effectuée.
Correspond à la variable Fortran NOMFOR (géré par DAMOCLES).
Type : une chaîne de caractères.
Valeurs possibles : toute chaîne de caractères correspondant à un fichier existant.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *FORTRAN FILE*.
Mots-clefs associés : FICHIER D'INITIALISATION
FICHIER DE MAILLAGE
FICHIER DES PARAMETRES

FICHIER SUITE

- *Mot-clef* ***obligatoire***.

Fichier de données pour une suite de calcul.
Correspond à la variable Fortran FICHSUITE.
Type : une chaîne de caractères.
Valeurs possibles : toute chaîne de caractères correspondant à un fichier existant.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *CONTINUATION FILE*.
Mots-clefs associés : FICHIER DES POSITIONS
FICHIER DES RESULTATS

GRAVITE

- *Mot-clef* ***obligatoire***.

Logique de choix de la gravité.
Correspond à la variable Fortran KGRAV.
Type : un logique.
Valeurs possibles : **OUI** (ou VRAI, ou YES, ou TRUE), **NON** (ou FAUX, ou NO, ou FALSE).
Valeur recommandée : aucune.
Valeur par défaut : OUI.
Version anglaise : *GRAVITY*.
Mot-clef associé : aucun.

LISSAGE DE LA DENSITE

- *Mot-clef* ***obligatoire***.

Logique de choix du lissage de la densité.

Correspond à la variable Fortran KLISS.
Type : un logique.
Valeurs possibles : **OUI** (ou VRAI, ou YES, ou TRUE), **NON** (ou FAUX, ou NO, ou FALSE).
Valeur recommandée : OUI.
Valeur par défaut : NON.
Version anglaise : *DENSITY SMOOTHING*.
Mot-clef associé : FACTEUR DE LISSAGE

MODELE DE DEFORMATION
Mot-clef facultatif, à utiliser uniquement si MODELE DE TURBULENCE n'est pas égal à 0.
Indice de choix pour le modèle de déformation.
Correspond à la variable Fortran KDEF.
Type : 1 entier.
Valeurs possibles : **1** (modèle standard, équations (141) à (143)),
2 (modèle de Violeau, équation (144)).
Valeur recommandée : 2.
Valeur par défaut : 1.
Version anglaise : *STRAIN MODEL*.
Mot-clef associé : MODELE DE TURBULENCE

MODELE DE GRADIENT DE PRESSION
- *Mot-clef* ***obligatoire***.

Indice de choix pour le modèle du gradient de pression.
Correspond à la variable Fortran KPRES.
Type : 1 entier.
Valeurs possibles : **1** (forme 1, équation (130)),
2 (forme 2, équation (131)).
Valeur recommandée : aucune.
Valeur par défaut : 2.
Version anglaise : *PRESSURE GRADIENT MODEL*.
Mot-clef associé : aucun.

MODELE DE PRODUCTION
Mot-clef facultatif, à utiliser uniquement si MODELE DE TURBULENCE est égal à 2 ou 3.
Indice de choix pour le modèle de production.
Correspond à la variable Fortran KPROD.
Valeurs possibles : **1** (modèle standard, équation (139)),
2 (modèle de Laurence, équation (140)).
Valeur recommandée : 2.
Valeur par défaut : 2.
Version anglaise : *PRODUCTION MODEL*.
Mot-clef associé : MODELE DE TURBULENCE

MODELE DE TURBULENCE
- *Mot-clef* ***obligatoire***.

Indice de choix pour le modèle de turbulence.
Correspond à la variable Fortran KTURB.
Type : 1 entier.
Valeurs possibles : **0** (viscosité turbulente constante, ou écoulement laminaire),
1 (modèle de longueur de mélange, équation (134)),
2 (modèle k–L_m, équations (136) et (137)),
3 (modèle k–ε, équations (136) et (138)).
Valeurs recommandées : 2 et 3.
Valeur par défaut : 2.
Version anglaise : *TURBULENCE MODEL*.
Mots-clefs associés : MODELE DE DEFORMATION
MODELE DE PRODUCTION
MODELE DE VITESSE DE FROTTEMENT
RAPPORT DELTA-DR
RAPPORT DEST-DR
RUGOSITE EQUIVALENTE

MODELE DE VISCOSITE

➢ *Mot-clef* ***obligatoire***.

Indice de choix pour le modèle du terme visqueux.
Correspond à la variable Fortran KVISQ.
Type : 1 entier.
Valeurs possibles : **1** (forme de Monaghan, équation (132)),
2 (forme de Morris, équation (133)).
Valeur recommandée : aucune.
Valeur par défaut : 1.
Version anglaise : *VISCOUS MODEL*.
Mot-clef associé : aucun.

MODELE DE VITESSE DE FROTTEMENT

Mot-clef facultatif, à utiliser uniquement si MODELE DE TURBULENCE n'est pas égal à 0.
Indice de choix pour le calcul de la vitesse de frottement.
Correspond à la variable Fortran KUSTAR.
Type : 1 entier.
Valeurs possibles : **1** (modèle logarithmique, équations (121) et (122)),
2 (modèle de gradient, équation (125)).
Valeur recommandée : 1.
Valeur par défaut : 2.
Version anglaise : *FRICTION VELOCITY MODEL*.
Mot-clef associé : MODELE DE TURBULENCE

MOTEUR

➢ *Mot-clef* ***obligatoire***.

Indice de choix du terme moteur selon x.
Correspond à la variable Fortran KMOT.

Type : 1 entier.
Valeurs possibles : **0** (pas de terme moteur),
1 (vitesse imposée en entrée),
2 (force motrice constante ajoutée à F^e dans l'équation (127)),
3 (force motrice ajustée, équation (156)).
Valeur recommandée : aucune. Dans le cas d'une force motrice, de préférence 3.
Valeur par défaut : 0.
Version anglaise : *DRIVING*.
Mots-clefs associés : VALEUR DE LA FORCE MOTRICE
VALEUR DE LA VITESSE DEBITANTE

NOMBRE DE FLUIDES

- *Mot-clef* ***obligatoire***.

Nombre de fluides impliqués dans la simulation.
Correspond à la variable Fortran NFLUID.
Type : 1 entier.
Valeurs possibles : **1**, **2**, **3** (NFLUIDMAX = 3).
Valeur recommandée : aucune.
Valeur par défaut : 1.
Version anglaise : *NUMBER OF FLUIDS*.
Mots-clefs associés : DENSITES DE REFERENCE
VISCOSITES MOLECULAIRES

NOMBRE DE PAS DE TEMPS

- *Mot-clef* ***obligatoire***.

Nombre d'itérations demandées.
Correspond à la variable Fortran NT.
Type : 1 entier.
Valeurs possibles : tous entiers supérieurs à 1.
Valeur recommandée : aucune.
Valeur par défaut : 1.
Version anglaise : *NUMBER OF TIME STEPS*.
Mots-clefs associés : NOMBRE DE PAS DE TEMPS
PERIODE POUR LES LISTINGS
PERIODE POUR LES SORTIES DES CHAMPS
PERIODE POUR LES SORTIES DES POSITIONS

PAROIS MOBILES

- *Mot-clef* ***obligatoire***.

Logique de choix pour la présence de parois mobiles.
Correspond à la variable Fortran KPARMOB.
Type : 1 logique.
Valeurs possibles : **OUI** (ou VRAI, ou YES, ou TRUE), **NON** (ou FAUX, ou NO, ou FALSE).
Valeur recommandée : aucune.
Valeur par défaut : NON.

Version anglaise : *MOVING WALLS*.
Mot-clef associé : aucun.

PERIODE DE SORTIE LISTING

- *Mot-clef* ***obligatoire***.

Nombre d'itérations séparant deux affichages à l'écran.
Correspond à la variable Fortran NLIST.
Type : 1 entier.
Valeurs possibles : tous entiers strictement positifs.
Valeur recommandée : aucune.
Valeur par défaut : 1.
Version anglaise : *LISTING PRINTOUT PERIOD*.
Mots-clefs associés : NOMBRE DE PAS DE TEMPS
PERIODE POUR LES SORTIES DES CHAMPS
PERIODE POUR LES SORTIES DES POSITIONS

PERIODE DE SORTIE DES CHAMPS

- *Mot-clef* ***obligatoire***.

Nombre d'itérations séparant deux impressions dans le fichier des résultats.
Correspond à la variable Fortran NSORTR.
Type : 1 entier.
Valeurs possibles : tous entiers strictement positifs.
Valeur recommandée : aucune.
Version anglaise : *FIELD PRINTOUT PERIOD*.
Mots-clefs associés : NOMBRE DE PAS DE TEMPS
PERIODE DE SORTIE LISTING
PERIODE POUR LES SORTIES DES POSITIONS

PERIODE DE SORTIE DES POSITIONS

- *Mot-clef* ***obligatoire***.

Nombre d'itérations séparant deux impressions dans le fichier des positions.
Correspond à la variable Fortran NSORTP.
Type : 1 entier.
Valeurs possibles : tous entiers strictement positifs.
Valeur recommandée : aucune.
Valeur par défaut : 1.
Version anglaise : *POSITION PRINTOUT PERIOD*.
Mots-clefs associés : PERIODE DE SORTIE LISTING
PERIODE POUR LES SORTIES DES CHAMPS

PERIODICITE SELON X

- *Mot-clef* ***obligatoire***.

Logique de choix pour un écoulement périodique selon x.
Correspond à la variable Fortran KPER.
Type : un logique.

Valeurs possibles : **OUI** (ou VRAI, ou YES, ou TRUE), **NON** (ou FAUX, ou NO, ou FALSE).
Valeur recommandée : aucune.
Valeur par défaut : NON.
Version anglaise : *X-PERIODICITY*.
Mot-clef associé : aucun.

POST-PROCESSEUR

- *Mot-clef **obligatoire***.

Indice de choix pour le post-processeur.
Correspond à la variable Fortran KVISU.
Type : un entier.
Valeurs possibles : **1** (Rubens©),
2 (Tecplot©).
Valeur recommandée : aucune.
Valeur par défaut : 1.
Version anglaise : *POSTPROCESSOR*.
Mot-clef associé : aucun.

RAPPORT DELTA-DR

Mot-clef facultatif, à utiliser uniquement si MODELE DE TURBULENCE n'est pas égal à 0.
Rapport $\delta / \delta r$.
Correspond à la variable Fortran DELTADR.
Type : 1 réel (sans unité).
Valeurs possibles : tous réels positifs, de préférence inférieurs à 1,0.
Valeurs recommandées : de l'ordre de grandeur de 0,1.
Valeur par défaut : 0,1.
Version anglaise : *RATIO DELTA-DR*.
Mot-clef associé : MODELE DE TURBULENCE

RAPPORT DEST-DR

Mot-clef facultatif, à utiliser uniquement si MODELE DE TURBULENCE n'est pas égal à 0.
Rapport $\delta_{est} / \delta r$.
Correspond à la variable Fortran DESTDR.
Type : 1 réel (sans unité).
Valeurs possibles : tous réels positifs, de préférence supérieurs à 1,0.
Valeurs recommandées : de l'ordre de grandeur de 1,0 à 4,0.
Valeur par défaut : 1,0.
Version anglaise : *RATIO DEST-DR*.
Mot-clef associé : MODELE DE TURBULENCE

RAPPORT H-DR

- *Mot-clef **obligatoire***.

Rapport $h / \delta r$.
Correspond à la variable Fortran HDR.
Type : un réel (sans unité).

Valeurs possibles : tous réels positifs (de préférence supérieurs à 1,0).
Valeur recommandée : de 1,2 à 1,5.
Valeur par défaut : 1,3.
Version anglaise : *RATIO H-DR*.
Mot-clef associé : aucun.

RUGOSITE EQUIVALENTE

➢ *Mot-clef* ***obligatoire***.

Rugosité équivalente des parois.
Correspond à la variable Fortran RUG.
Type : un réel (m).
Valeurs possibles : tous réels positifs.
Valeurs recommandées : de préférence inférieures à δr.
Valeur par défaut : 10^{-4} m.
Version anglaise : *EQUIVALENT ROUGHNESS*.
Mot-clef associé : aucun.

SUITE DE CALCUL

➢ *Mot-clef* ***obligatoire***.

Logique de choix pour une suite de calcul.
Correspond à la variable Fortran KSUITE.
Type : un logique.
Valeurs possibles : **OUI** (ou VRAI, ou YES, ou TRUE), **NON** (ou FAUX, ou NO, ou FALSE).
Valeur recommandée : aucune.
Valeur par défaut : NON.
Version anglaise : *CALCULATION CONTINUED*.
Mot-clef associé : aucun.

TYPE DE CONDITION DE PAROIS

➢ *Mot-clef* ***obligatoire***.

Indice de choix pour la modélisation des parois.
Correspond à la variable Fortran KCPAR.
Type : 1 entier.
Valeurs possibles : **1** (particules fictives, **§ 1.1.4.3**),
2 (forces de parois, équations (105) à (109)).
Valeur recommandée : 1.
Valeur par défaut : 1.
Version anglaise : *WALL TREATMENT*.
Mots-clefs associés : COEFFICIENT DES FORCES DE PAROI
TYPE DE FORCE DE PAROI

TYPE DE FORCES DE PAROI

Mot-clef facultatif, à utiliser uniquement si TYPE DE CONDITIONS AUX PAROIS vaut 2.
Choix du modèle de forces de paroi.

Correspond à la variable Fortran KFPAR.
Type : 1 entier.
Valeurs possibles : **1** (forme purement répulsive, équation (106)),
2 (forme de Monaghan, équation (107)),
3 (forme de Lennard-Jones, équations (108) et (109)).
Valeur recommandée : 1.
Version anglaise : *WALL FORCE TYPE*.
Valeur par défaut : 1.
Mots-clefs associés : COEFFICIENT DES FORCES DE PAROI
TYPE DE CONDITIONS AUX PAROIS

TYPE DU NOYAU

- *Mot-clef* ***obligatoire***.

Type de noyau w_h.
Correspond à la variable Fortran KKERNEL.
Type : 1 entier.
Valeurs possibles : **1** (ordre 3, équation (13)),
2 (ordre 4, équation (14)),
3 (ordre 5, équation (15)).
Valeur recommandée : 2.
Valeur par défaut : 2.
Version anglaise : *KERNEL TYPE*.
Mot-clef associé : aucun.

VALEUR DE LA FORCE MOTRICE

Mot-clef facultatif, utiliser uniquement si MOTEUR vaut 2.
Force motrice horizontale.
Correspond à la variable Fortran FORCST.
Type : un réel ($m.s^{-2}$).
Valeurs possibles : tous réels.
Valeur recommandée : aucune.
Valeur par défaut : 0,05 $m.s^{-2}$.
Version anglaise : *DRIVING FORCE VALUE*.
Mot-clef associé : MOTEUR

VALEUR DE LA VITESSE DEBITANTE

Mot-clef facultatif, à utiliser uniquement si MOTEUR vaut 3.
Vitesse débitante imposée.
Correspond à la variable Fortran DEBREF.
Type : un réel ($m.s^{-1}$).
Valeurs possibles : tous réels.
Valeur recommandée : aucune.
Valeur par défaut : 1,0 $m.s^{-1}$.
Version anglaise : *BULK VELOCITY VALUE*.
Mot-clef associé : MOTEUR

VITESSE DU SON

VISCOSITES MOLECULAIRES

- *Mot-clef* ***obligatoire***.

Tableau des viscosités moléculaires des fluides considérés.
Correspond à la variable Fortran NU0.
Type : NFLUID réels ($m^2.s^{-1}$) séparés par des virgules.
Valeurs possibles : tous réels strictement positifs.
Valeur recommandée : aucune.
Valeur par défaut : aucune.
Version anglaise : *MOLECULAR VISCOSITIES*.
Mot-clef associé : DENSITES DE REFERENCE
NOMBRE DE FLUIDES

VITESSE DU SON

- *Mot-clef* ***obligatoire***.

Vitesse du son numérique.
Correspond à la variable Fortran VITC0.
Type : un réel ($m.s^{-1}$).
Valeurs possibles : tous réels strictement positifs.
Valeur recommandée : au moins 10 fois $\max\left(u_{\max}, \sqrt{gH_{\max}}\right)$.
Valeur par défaut : 100 $m.s^{-1}$.
Version anglaise : *SPEED OF SOUND*.
Mot-clef associé : aucun.

2.3.3 Liste des principales variables FORTRAN

Les *principales* variables de SPARTACUS-2D sont listées ci-après dans l'ordre alphabétique. Certaines variables locales, par exemple, ont été omises. Pour chacune des variables sont indiquées (s'il y a lieu) déclaration, unité, fichiers FORTRAN concernés, définition, traduction anglaise, variable correspondante dans les équations du modèle (Cf. 1[e] partie), valeur (pour les constantes) et mot-clef correspondant, au paragraphe précédent (le cas échéant, indiqué par une flèche).

AX, AZ DOUBLE PRECISION AX(NPMAX), AZ(NPMAX) $m.s^{-2}$
Calculés dans impulsion.f, utilisés dans mouvement.f et pastemps.f.
Composantes des accélérations des particules.
Version anglaise : *Particle acceleration components.*
Correspondent au vecteur $d\underline{u}_a / dt$ dans le système (145).

CK1 DOUBLE PRECISION CK1 Sans unité
Défini dans initial.f, utilisé dans kequation.f et entpar.f.
L'un des coefficient du modèle k–ε.
Version anglaise : *One of the k–ε model coefficients*.
Correspond au coefficient $C_\mu^{3/4}$ de l'équation (90).
$C_\mu^{3/4} = 0{,}16$.

CMU DOUBLE PRECISION CMU Sans unité
Défini dans initial.f, utilisé dans kequation.f et entpar.f.

L'un des coefficient du modèle k–ε.
Version anglaise : *One of the k–ε model coefficients*.
Correspond au coefficient C_μ de l'équation (76).
$C_\mu = 0{,}09$.

CPAR DOUBLE PRECISION CPAR — $m^2.s^{-2}$
Lu dans lecparam.f, utilisé dans forcparois.f.
Coefficient de la force de parois n° 1.
Version anglaise : *Wall force 1 coefficient*.
Correspond au paramètre C de l'équation (106).
➢ Mot-clef COEFFICIENT DES FORCES DE PAROIS.

CR DOUBLE PRECISION CR — Sans unité
Défini dans initial.f, utilisé dans frottement.f et mouvement.f.
Constante additive de la loi logarithmique rugueuse.
Version anglaise : *Coefficient relative to a rough log law.*
Correspond au paramètre C_* de l'équation (112).
$C_* = 8{,}5$.

CT INTEGER CT — Sans unité
Initialisé dans initial.f, utilisé dans ecrit.f.
Compteur des zones pour Tecplot©.
Version anglaise : Tecplot© *current zone number*.

CTHETA, STHETA DOUBLE PRECISION CTHETA(NBMAX), STHETA(NBMAX) — Sans unité
calculés dans lecinit.f ou lecsuit.f, utilisés dans frottement.f initial.f et mouvement.f.
Composantes des vecteurs normaux aux parois.
Version anglaise : *Components of wall normal vectors*.
Correspondent aux composantes du vecteur $\underline{n}_a$ dans l'équation (123).

DEB DOUBLE PRECISION DEB — $m.s^{-1}$
Défini et utilisé dans forcext.f.
Vitesse débitante calculée.
Version anglaise : *Computed mean bulk velocity*.

DEBREF DOUBLE PRECISION DEBREF — $m.s^{-1}$
Lu dans lecparam.f, utilisé dans forcext.f.
Vitesse débitante imposée.
Version anglaise : *Prescribed mean bulk velocity*.
➢ Mot-clef VALEUR DE LA VITESSE DEBITANTE.

DELTA DOUBLE PRECISION DELTA — m
Défini dans initial.f, utilisé dans frottement.f, kequation.f, epsilon.f, melange.f, et mouvement.f.
Distance entre particules de bord et bords réels.
Version anglaise : *Edge particle distance to real walls*.
Correspond au paramètre δ de l'équation (112).

DELTADR DOUBLE PRECISION DELTADR — Sans unité
Lu dans lecparam.f, utilisé dans initial.f.
Rapport δ / δr.
Version anglaise : *Ratio* δ / δr.
- Mot-clef RAPPORT DELTA-DR.

DEST DOUBLE PRECISION DEST — m
Calculé dans frottement.f.
Distance d'estimation pour la loi de paroi
Version anglaise : *Estimation length for the wall function*.
Correspond à la variable δ_{est} de l'équation (121).

DESTDR DOUBLE PRECISION DESTDR — Sans unité
Lu dans lecparam.f, utilisé dans frottement.f.
Rapport δ_{est} / δr.
Version anglaise : *Ratio* δ_{est} / δr.
- Mot-clef RAPPORT DEST-DR.

DR DOUBLE PRECISION DR — m
Lu dans lecinit.f ou lecsuit.f, utilisé dans ouverture.f, frottement.f, forcext.f, initial.f, melange.f et entpar.f.
Distance initiale entre deux particules voisines.
Version anglaise : *Initial interparticle spacing.*
Correspond au paramètre δr.

DT DOUBLE PRECISION DT — s
Calculé dans pastemps.f, utilisé dans initial.f, mouvement.f, continuite.f, kequation.f, forcext.f et epsilon .f.
Pas de temps.
Version anglaise : *Time step*.
Correspond au paramètre δt.

EPS DOUBLE PRECISION EPS(NPMAX) — $m^2.s^{-3}$
Initialisé dans initial.f, calculé dans epsilon.f, utilisé dans interpol.f, ecrit.f, kequation.f, epsilon.f, entpar.f et sorpar.f.
Taux de dissipation d'énergie.
Version anglaise : *Dissipation rate.*
Correspond à la quantité ε_a de l'équation (138).

ETA2 DOUBLE PRECISION ETA2 — m^2
Calculé dans initial.f, utilisé dans visco.f, tauxdef.f et kequation.f.
Terme de correction pour la diffusion.
Version anglaise : *Correction term for diffusion.*
Correspond au paramètre η^2 (Cf. remarque au dessous de l'équation (25)).
$\eta^2 = 0{,}01h^2$

FACTLISS DOUBLE PRECISION FACTLISS — Sans unité
Lu dans lecparam.f, utilisé dans spartacus2d.f et continuite.f.

Facteur de lissage de la densité.
Version anglaise : *Density smoothing factor*.
Correspond au paramètre ε_ρ (équation (148)).

FEXTX, FEXTZ DOUBLE PRECISION FEXTX(NPMAX) — $m.s^{-2}$
Définis dans forcext.f, utilisé dans impulsion.f.
Composantes des forces extérieures.
Version anglaise : *External force components*.
Correspondent aux composantes du vecteur $\underline{F}_a^e$ dans l'équation (127).

FICHINIT CHARACTER*80 FICHINIT — Sans unité
Lu dans lecparam.f, utilisé dans initial.f et lecinit.f.
Fichier d'initialisation.
Version anglaise : *Initializing file*.
- Mot-clef FICHIER D'INITIALISATION.

FICHMAILL CHARACTER*80 FICHMAILL — Sans unité
Défini dans spartacus2d.f, utilisé dans initial.f et ecrit.f.
Fichier maillage pour Rubens©.
Version anglaise : *Mesh file for* Rubens©.
- Mot-clef FICHIER DE MAILLAGE.

FICHPARAM CHARACTER*40 FICHPARAM — Sans unité
Défini dans spartacus2d.f, utilisé dans initial.f et lecparam.f.
Fichier des paramètres.
Version anglaise : *Steering file*.
- Mot-clef FICHIER DES PARAMETRES.

FICHPOS CHARACTER*40 FICHPOS — Sans unité
Défini dans spartacus2d.f, utilisé dans ouverture.f et ecrit.f.
Fichier des positions.
Version anglaise : *Position file*.
- Mot-clef FICHIER DES POSITIONS.

FICHRES CHARACTER*40 FICHRES — Sans unité
Défini dans spartacus2d.f, utilisé dans ouverture.f et ecrit.f.
Fichier des champs continus.
Version anglaise : *Continuous fields file*.
- Mot-clef FICHIER DES RESULTATS.

FICHSUITE CHARACTER*40 FICHSUITE — Sans unité
Défini dans spartacus2d.f, utilisé dans initial.f et lecsuit.f.
Fichier de données pour une suite de calcul.
Version anglaise : *Data file for computation continued*.
- Mot-clef FICHIER SUITE.

FMOT DOUBLE PRECISION FMOT(NPMAX) — $m.s^{-2}$
Défini et utilisé dans forcext.f.
Force motrice horizontale.

Version anglaise : *Axial driving force*.

FORCST DOUBLE PRECISION FORCST — $m.s^{-2}$
Lu dans lecparam.f, utilisé dans forcext.f.
Force motrice horizontale imposée.
Version anglaise : *Imposed axial driving force*.
➢ Mot-clef VALEUR DE LA FORCE MOTRICE.

FPARX, FPARZ DOUBLE PRECISION FPARX(NPMAX), FPARZ(NPMAX) — $m.s^{-2}$
Calculé dans forcparois.f, utilisé dans forcext.f, impulsion.f et sorpar.f.
Composantes des forces de parois.
Version anglaise : *Wall force components*.
Correspondent aux composantes de la somme sur b des $\underline{F}^{p}_{b\to a}$ dans l'équation (105).

FPRES DOUBLE PRECISION FPRES(NPMAX,NLIENMAX) — $kg^{-1}.m^{5}.s^{-2}$
Calculé dans fluxpres.f, utilisé dans impulsion.f
Flux d'impulsion dû aux forces de pression.
Version anglaise : *Momentum flux relative to pressure forces.*
Correspond à la quantité $\overline{\underline{q}}_{p,ba} / \nabla_a w_h(r_{ab})$ de l'équation (127).

FVISQ DOUBLE PRECISION FVISQ(NPMAX,NLIENMAX) — $kg^{-1}.m^{5}.s^{-2}$ ou $kg^{-1}.s^{-1}$
Calculé dans fluxvisq.f, utilisé dans impulsion.f
Flux d'impulsion dû aux forces visqueuses.
Version anglaise : *Momentum flux relative to viscous forces.*
Correspond à la quantité $\overline{\underline{q}}_{v,ba} / \nabla_a w_h(r_{ab})$ (modèle visqueux de Monaghan) ou $\overline{\underline{q}}_{v,ba} / \overline{\underline{u}}_{ab}$ (Modèle de Morris) de l'équation (127).

GAMMAX, GAMMAZ DOUBLE PRECISION GAMMAX, GAMMAZ — s^{-1}
Lus dans lecparam.f, utilisés dans impulsion.f.
Coefficients d'amortissement.
Version anglaise : *Damping coefficients*.
Correspondent aux paramètres Γ_x et Γ_z de l'équation (157), p. 100.
➢ Mots-clef COEFFICIENT D'AMORTISSEMENT SELON X et COEFFICIENT D'AMORTISSEMENT SELON Z.

GKERX, GKERZ DOUBLE PRECISION GKERX(NPMAX), GKERZ(NPMAX) — m^{-3}
Calculés dans kernel.f, utilisés dans gradvit.f, impulsion.f, kequation.f, fluxvisq.f, tauxdef.f et continuite.f.
Composantes du gradient du noyau.
Version anglaise : *Kernel derivative components*.
Correspondent aux composantes du vecteur $\nabla_a w_h(r_{ab})$ dans (126).

GRAV DOUBLE PRECISION GRAV — $m.s^{-2}$
Défini dans initial.f, utilisé dans forcext.f.
Gravité.
Version anglaise : *Gravity*.

Correspond au paramètre g.
g = 9,807 m.s^{-2}.

GVXX, GVXZ, GVZX, GVZZ DOUBLE PRECISION GVXX(NPMAX), etc. — s^{-1}
Calculés dans gradvit.f, utilisés dans kernel.f gradvit.f, frottement.f et tauxdef.f.
Composantes du tenseur gradient de vitesse.
Version anglaise : C*omponents of the velocity gradient tensor*.
Correspondent aux composantes du vecteur $\nabla \underline{u}$ dans l'équation (65).

H DOUBLE PRECISION H — m
Calculé dans initial.f, utilisé dans kernel.f, pastemps.f, tablien.f, interpol.f, frottement.f, forcparois.f et continuite.f.
Longueur d'atténuation.
Version anglaise : *Smoothing length*.
Correspond au paramètre h.

HDR DOUBLE PRECISION HDR — Sans unité
Lu dans lecparam.f, utilisé dans initial.f.
Rapport h / δr.
Version anglaise : *Ratio h* / δ*r*.
➢ Mot-clef RAPPORT H-DR.

I, J, L INTEGER I, J, L — Sans unité
Définis et utilisé dans toutes les routines.
Indices de particules. I est généralement la particule considérée dans l'instant, J un indice allant de 1 à NLIEN(I), et L = ILIEN(I,J) l'indice de la J-ième particule liée à I (voir variables NLIEN et ILIEN).

ILIEN INTEGER ILIEN(NPMAX,NLIENMAX) — Sans unité
Défini dans tablien.f, utilisé dans kernel.f, gradvit.f, impulsion.f, forcparois.f, viscturb.f, kequation.f, epsilon.f, fluxpres.f, fluxvisq.f, tauxdef.f et continuite.f.
Tableau de liens des particules.
Version anglaise : *Particle link list*.

IPARTQ INTEGER IPARTQ(NQUADXMAX, NQUADZMAX,NPARTQMAX) — Sans unité
Défini dans tablien.f, utilisé dans kernel.f, interpol.f et frottement.f.
Indice des particules situées dans un carré.
Version anglaise : *Index of particles located in a square*.

IT INTEGER IT — Sans unité
Défini dans spartacus2d.f et utilisé dans ecrit.f, impulsion.f, forcext.f, entpar.f et sorpar.f.
Indice de l'itération en cours.
Version anglaise : *Index of the current iteration*.

KAPPA DOUBLE PRECISION KAPPA — Sans unité
Défini dans initial.f, utilisé dans kequation.f, epsilon.f, frottement.f et mouvement.f.
Constante de von Karman.
Version anglaise : *Von Karman constant.*
Correspond au paramètre κ de l'équation (112).
$\kappa = 0,41$.

KCPAR INTEGER KCPAR — Sans unité
Lu dans lecparam.f, utilisé dans forcparois.f, fluxpres.f et forcext.f.
Indice de choix pour la modélisation des parois.
Version anglaise : *Choice index for wall modelling.*
- Mot-clef TYPE DE CONDITION DE PAROI.

KDEF INTEGER KDEF — Sans unité
Lu dans lecparam.f, utilisé dans tauxdef.f.
Indice de choix pour le modèle de taux de déformation.
Version anglaise : *Choice index for rate-of-strain model.*
- Mot-clef MODELE DE DEFORMATION.

KENT INTEGER KENT(NPMAX) — Sans unité
Lu dans lecinit.f ou lecsuit.f, utilisé dans entpar.f, sorpar.f et ecrit.f.
Type de condition aux frontières fluides.
Version anglaise : *Type of condition on fluid boundaries*.

KER4_VAL DOUBLE PRECISION KER4_VAL(NPMAX,0:NLIENMAX) — m^{-2}
Calculé dans kernel4_val.f, utilisé dans spartacus2d.f et continuite.f.
Valeur du noyau d'ordre 4.
Version anglaise : *Fourth order kernel value*.

KFLUID INTEGER KFLUID(NPMAX) — Sans unité
Lu dans lecinit.f ou lecsuit.f, utilisé dans etat.f, continuite.f, etat.f, fluxvisq.f, kequation.f, epsilon.f, pastemps.f, entpar.f et sorpar.f.
Type de fluide.
Version anglaise : *Fluid type*.

KFPAR INTEGER KFPAR — Sans unité
Lu dans lecparam.f, utilisé dans forcparois.f.
Indice de choix pour les forces de parois.
Version anglaise : *Choice index for wall forces*.
- Mot-clef TYPE DE FORCES DE PAROI.

KGAMMAX, KGAMMAZ LOGICAL KGAMMAX, KGAMMAZ — s^{-1}
Lus dans lecparam.f, utilisés dans impulsion.f.
Logiques de choix d'amortissement.

Version anglaise : *Choice logical indices for damping*.
- Mots-clef AMORTISSEMENT SELON X et AMORTISSEMENT SELON Z.

KGRAV LOGICAL KGRAV — Sans unité
Lu dans lecparam.f, utilisé dans forcext.f et continuite.f.
Logique de choix de la gravité.
Version anglaise : *Logical index for gravity choice*.
- Mot-clef GRAVITE.

KKERNEL INTEGER KKERNEL — Sans unité
Lu dans lecparam.f, utilisé dans kernel.f et initial.f.
Indice de choix du noyau.
Version anglaise : *Choice index for kernel*.
- Mot-clef TYPE DU NOYAU.

KLISS LOGICAL KLISS — Sans unité
Lu dans lecparam.f, utilisé dans spartacus2d.f et continuite.f.
Logique de choix du lissage de la densité.
Version anglaise : *Logical relative to density smoothing.*

KLIST LOGICAL KLIST — Sans unité
Défini dans spartacus2d.f et utilisé dans tablien.f, entpar.f, sorpar.f, pastemps.f et pastemps.f.
Logique de listing.
Version anglaise : *Listing logical.*

KMOT INTEGER KMOT — Sans unité
Lu dans lecparam.f, utilisé dans forcext.f.
Indice de choix du terme moteur.
Version anglaise : *Choice index for the forcing term*.
- Mot-clef MOTEUR.

KPAR INTEGER KPAR(NPMAX) — Sans unité
Lu dans lecinit.f ou lecsuit.f, utilisé dans gradvit.f, fluxpres.f, mouvement.f, ecrit.f, forcparois.f, melange.f, entpar.f, kequation.f et sorpar.f.
Type de particule
Version anglaise : *Particle type*.

KPARM LOGICAL KPARM — Sans unité
Lu dans lecparam, utilisé dans mouvement.f.
Logique de choix de parois mobiles.
Version anglaise : *Logical index for moving walls*.
- Mot-clef PAROIS MOBILES.

KPARMOB INTEGER KPARMOB(NPMAX) — Sans unité
Lu dans lecinit.f ou lecsuit.f, utilisé dans mouvement.f et paroimobile.f.

Type de paroi (mobile ou non).
Version anglaise : *Moving wall or edge particle type*.

KPER LOGICAL KPER — Sans unité
Lu dans lecparam.f, utilisé dans tablien.f, interpol.f, frottement.f, mouvement.f et sorpar.f.
Logique de choix de périodicité.
Version anglaise : *Logical choice for periodicity*.
- Mot-clef PERIODICITE SELON X.

KPRES INTEGER KPRES — Sans unité
Lu dans lecparam.f, utilisé dans fluxpres.f.
Indice de choix pour le modèle du gradient de pression.
Version anglaise : *Choice index for the pressure gradient model.*
- Mot-clef MODELE DE GRADIENT DE PRESSION.

KPROD INTEGER KPROD — Sans unité
Lu dans lecparam.f, utilisé dans kequation.f.
Indice de choix pour le modèle de production.
Version anglaise : *Choice index for production model.*
- Mot-clef MODELE DE PRODUCTION.

KSORTP LOGICAL KSORTP — Sans unité
Calculé et utilisé dans spartacus2d.f et ecrit.f.
Logique pour les sorties positions.
Version anglaise : *Logical for position output.*

KSORTR LOGICAL KSORTR — Sans unité
Calculé et utilisé dans spartacus2d.f et ecrit.f.
Logique pour les sorties des champs continus.
Version anglaise : *Logical for field output*.

KSUITE LOGICAL KSUITE — Sans unité
Lu dans lecparam.f, utilisé dans initial.f.
Logique de suite de calcul.
Version anglaise : *Logical for calculation continued*.
- Mot-clef SUITE DE CALCUL.

KTURB INTEGER KTURB — Sans unité
Lu dans lecparam.f, utilisé dans initial.f, forcext.f, fluxpres.f, impulsion.f, viscturb.f, fluxvisq.f, kequation.f, entpar.f et mouvement.f.
Indice de choix pour le modèle de turbulence.
Version anglaise : *Choice index for the turbulence model.*
- Mot-clef MODELE DE TURBULENCE.

KUSTAR INTEGER KUSTAR — Sans unité
Lu dans lecparam.f, utilisé dans frottement.f.
Indice de choix pour le calcul de la vitesse de frottement.

Version anglaise : *Choice index for the friction velocity computation.*

➢ Mot-clef MODELE DE VITESSE DE FROTTEMENT.

KVISQ INTEGER KVISQ — Sans unité

Lu dans lecparam.f, utilisé dans fluxvisq.f et impulsion.f.
Indice de choix pour le modèle du terme visqueux.
Version anglaise : *Choice index for the viscous term model.*

➢ Mot-clef MODELE DE VISCOSITE.

KVISU INTEGER KVISU — Sans unité

Lu dans lecparam.f, utilisé dans ouverture.f et ecrit.f.
Indice de choix pour le post-processeur.
Version anglaise : *Choice index the postprocessor.*

➢ Mot-clef POST-PROCESSEUR.

LM DOUBLE PRECISION LM(NPMAX) — m

Défini dans melange.f, utilisé dans viscturb.f, entpar.f, initial.f et sorpar.f
Longueur de mélange.
Version anglaise : *Mixing length.*
Correspond à la quantité $L_{m,a}$ de l'équation (81).

MASS DOUBLE PRECISION MASS(NPMAX) — kg

Défini dans initial.f, utilisé dans interpol.f, ecrit.f, gradvit.f, frottement.f, impulsion.f, continuite.f, entpar.f, epsilon.f, kequation.f, lecsuit.f, privee.f, tauxdef.f et sorpar.f.
Masse des particules.
Version anglaise : *Particle mass.*
Correspond à la variable m_b dans le système (126).

NBMAX INTEGER NBMAX — Sans unité

Défini dans spartacus2d.f, utilisé dans initial.f, lecinit.f, lecsuit.f, ecrit.f, frottement.f, kequation.f, epsilon.f et mouvement.f.
Nombre maximum de particules de bord.
Version anglaise : *Maximum number of edge particles.*

NFLUID INTEGER NFLUID — Sans unité

Lu dans lecparam.f.
Nombre de fluides.
Version anglaise : *Number of fluids.*

➢ Mot-clef NOMBRE DE FLUIDES.

NFLUIDMAX INTEGER NFLUIDMAX — Sans unité

Défini dans spartacus2d.f, utilisé dans continuite.f, etat.f, fluxvisq.f, kequation.f, epsilon.f, entpar.f, viscturb.f et pastemps.f.
Nombre maximum de fluides.
Version anglaise : *Maximum number of fluids.*

NFLUIDMAX = 3.

NLIEN INTEGER NLIEN(NPMAX) Sans unité
Défini dans spartacus2d.f, utilisé dans tablien.f, kernel.f, gradvit.f, impulsion.f, forcparois.f, fluxpres.f, fluxvisq.f, tauxdef.f, kequation.f, epsilon.f et continuite.f.
Nombre de liens d'une particule.
Version anglaise : *Number of links relative to a particle.*

NLIENMAX INTEGER NLIENMAX Sans unité
Défini dans spartacus2d.f, utilisé dans tablien.f, kernel.f, gradvit.f, impulsion.f, forcparois.f, fluxpres.f, fluxvisq.f, tauxdef.f, kequation.f, epsilon.f et continuite.f.
Nombre de liens maximum d'une particule.
Version anglaise : *Maximum number of links relative to a particle.*

NLIST INTEGER NLIST Sans unité
Lu dans lecparam.f, utilisé dans spartacus2d.f.
Période de sorties listing.
Version anglaise : *Listing printout period.*
- Mot-clef PERIODE DE SORTIES LISTING.

NPARB INTEGER NPARB Sans unité
Défini dans lecinit.f ou lecsuit.f, utilisé dans ecrit.f, gradvit.f, frottement.f, tauxdef.f, kequation.f, epsilon.f, mouvement.f et continuite.f.
Nombre de particules de bord.
Version anglaise : *Number of edge particles.*

NPARF INTEGER NPARF Sans unité
Défini dans lecinit.f ou lecsuit.f, utilisé dans ecrit.f, gradvit.f, frottement.f, tauxdef.f, kequation.f, epsilon.f, pastemps.f mouvement.f, continuite.f, entpar.f et sorpar.f.
Nombre de particules fluides.
Version anglaise : *Number of fluid particles.*

NPART INTEGER NPART Sans unité
Défini dans lecinit.f ou lecsuit.f, utilisé dans tablien.f, kernel.f, mouvement.f, continuite.f, etat.f, entpar.f et sorpar.f.
Nombre total de particules.
Version anglaise : *Total particle number.*

NPARTQ INTEGER NPARTQ(NQUADXMAX,NQUADZMAX) Sans unité
Calculé dans tablien.f, utilisé dans interpol.f et frottement.f.
Nombre de particules dans un carré.
Version anglaise : *Number of particles in a square.*

NPARTQMAX INTEGER NPARTQMAX Sans unité

Calculé dans tablien.f, utilisé dans interpol.f
et frottement.f.
Nombre maximum de particules dans un carré.
Version anglaise : *Maximum number of particles in a square.*

NPMAX INTEGER NPMAX Sans unité
Défini dans spartacus2d.f, utilisé dans toutes les routines.
Nombre maximum de particules.
Version anglaise : *Maximum number of particles.*

NQUADX INTEGER NQUADX Sans unité
Calculé dans tablien.f, utilisé dans interpol.f
et frottement.f.
Nombre de carrés sur l'axe *x*.
Version anglaise : *Number of squares along x.*

NQUADXMAX INTEGER NQUADXMAX Sans unité
Calculé dans tablien.f, utilisé dans interpol.f
et frottement.f.
Nombre maximum de carrés sur l'axe *x*.
Version anglaise : *Maximum number of squares along x.*

NQUADZ INTEGER NQUADZ Sans unité
Calculé dans tablien.f, utilisé dans interpol.f
et frottement.f.
Nombre de carrés sur l'axe *z*.
Version anglaise : *Number of squares along z.*

NQUADZMAX INTEGER NQUADZMAX Sans unité
Calculé dans tablien.f, utilisé dans interpol.f
et frottement.f.
Nombre maximum de carrés sur l'axe *z*.
Version anglaise : *Maximum number of squares along z.*

NSORTP INTEGER NSORTP Sans unité
Lu dans lecparam.f, utilisé dans spartacus2d.f.
Période pour l'écriture des positions des particules.
Version anglaise : *Position output period.*
➢ Mot-clef PERIODE POUR LES SORTIES DES POSITIONS.

NSORTR INTEGER NSORTR Sans unité
Lu dans lecparam.f, utilisé dans spartacus2d.f.
Période pour l'écriture des champs continus.
Version anglaise : *Field output period.*
➢ Mot-clef PERIODE POUR LES SORTIES DES CHAMPS.

NT INTEGER NT Sans unité
Lu dans lecparam.f, utilisé dans spartacus2d.f.
Nombre de pas de temps.

Version anglaise : *Number of time steps.*

- Mot-clef NOMBRE DE PAS DE TEMPS.

NU0 INTEGER NU0(NFLUIDMAX) $m^2.s^{-1}$
Lu dans lecparam.f, utilisé dans fluxvisq.f, kequation.f, epsilon.f et pastemps.f.
Viscosités moléculaires.
Version anglaise : *Molecular viscosities.*
Correspond au paramètre ν de l'équation (67).

- Mot-clef VISCOSITES MOLECULAIRES.

NUT DOUBLE PRECISION NUT(NPMAX) $m^2.s^{-1}$
Initialisé dans initial.f, calculé dans viscturb.f, utilisé dans interpol.f, ecrit.f, fluxvisq.f, kequation.f, epsilon.f, pastemps.f, entpar.f et sorpar.f.
Viscosité turbulente.
Version anglaise : *Eddy viscosity.*
Correspond à ν_T dans l'équation (67), ou à $\nu_{T,a} - \nu$ dans (132) et (133).

P DOUBLE PRECISION P(NPMAX) $kg.\ m^{-1}.s^{-2}$
Calculé dans etat.f, utilisé dans interpol.f, ecrit.f, fluxpres.f, entpar et sorpar.f.
Pression.
Version anglaise : *Pressure*.
Correspond à la variable p_a de l'équation (131).

PI DOUBLE PRECISION PI Sans unité
Défini dans initial.f, utilisé dans interpol.f, kernel.f et frottement.f.
Nombre d'Archimède.
Version anglaise : *Archimede's number*.
$\pi = 3.141592653589793$.

PRIV DOUBLE PRECISION PRIVE(NPMAX) ?
Créé dans privee.f, utilisé dans ecrit.f.
Tableau de sortie privé pour l'utilisateur.
Version anglaise : *Private printout variable for the user.*

PROD DOUBLE PRECISION PROD(NPMAX) $m^2.s^{-3}$
Calculé dans kequation.f, utilisé dans epsilon.f.
Taux de production d'énergie.
Version anglaise : *Production rate.*
Correspond à la quantité P_a de l'équation (136).

R0 DOUBLE PRECISION R0 m
Calculé dans initial.f, utilisé dans forcparois.f.
Rayon d'action des parois.
Version anglaise : *Wall action distance.*

Correspond au paramètre r_0 intervenant dans le **§ 1.1.4.2**.

RAB DOUBLE PRECISION RAB(NPMAX,NLIENMAX) m
Calculé dans tablien.f, utilisé dans kernel.f, impulsion.f, forcparois.f, fluxvisq.f, continuite.f et kequation.f.
Distances entre particules.
Version anglaise : *Particle distance.*
Correspond au paramètre r_{ab} de l'équation (126).

RHO DOUBLE PRECISION RHO(NPMAX) $kg.m^{-2}$
Calculé dans continuite.f, utilisé dans interpol.f, ecrit.f, frottement.f, etat.f, entpar.f et sorpar.f.
Masse volumique.
Version anglaise : *Density*.
Correspond au paramètre ρ_a de l'équation (126).

RHO0 DOUBLE PRECISION RHO0(NFLUID) $kg.m^{-2}$
Lu dans lecparam.f, utilisé dans etat.f, continuite.f et entpar.f.
Densités de référence.
Version anglaise : *Reference densities.*
Correspond au paramètre ρ_0 de l'équation (129).
➢ Mot-clef DENSITES DE REFERENCE.

RUG DOUBLE PRECISION RUG m
Lu dans lecparam.f, utilisé dans frottement.f et mouvement.f.
Rugosité des parois.
Version anglaise : *Wall roughness.*
Correspond au paramètre k_s de l'équation (112).
➢ Mot-clef RUGOSITE EQUIVALENTE.

S DOUBLE PRECISION S(NPMAX) s^{-1}
Calculé dans tauxdef.f, utilisé dans kequation.f, epsilon.f, interpol.f, ecrit.f, entpar.f, initial.f et sorpar.f.
Taux de déformation.
Version anglaise : *Rate-of-strain.*
Correspond à la quantité S_a de l'équation (135).

SUPP DOUBLE PRECISION SUPP Sans unité
Défini dans initial.f, utilisé dans interpol.f, tablien.f et frottement.f.
Support du noyau.
Version anglaise : *Kernel support.*
Correspond au rapport α_t du **§ 1.1.1.1**.

TEMPS DOUBLE PRECISION TEMPS s
Calculé dans spartacus2d.f, utilisé dans initial.f, interpol.f, paroimobile.f, pastemps.f et ecrit.f.
Temps physique.

Version anglaise : *Physical time.*
Correspond à la grandeur t.

THETA DOUBLE PRECISION THETA(NBMAX) — °
Lu dans lecinit.f ou lecsuit.f, utilisé dans ecrit.f.
Angle entre le vecteur normal intérieur à la paroi et l'axe x.
Version anglaise : *Angle between wall normal vector and x-axis*.
Correspond à l'angle de la **figure 15**, p. 101.

TKE DOUBLE PRECISION TKE(NPMAX) — $m^2.s^{-2}$
Initialisé dans initial.f, calculé dans kequation.f, utilisé dans interpol.f, ecrit.f, fluxpres.f, kequation.f, epsilon.f, entpar.f et sorpar.f.
Energie cinétique turbulente.
Version anglaise : *Turbulent kinetic energy.*
Correspond à la quantité k_a de l'équation (136).

USTAR DOUBLE PRECISION USTAR(NBMAX) — $m.s^{-1}$
Calculé dans frottement.f, utilisé dans kequation.f, epsilon.f et mouvement.
Vitesse de frottement.
Version anglaise : *Friction velocity*.
- Correspond au paramètre $u_{*,a}$ de l'équation (115).

VEST DOUBLE PRECISION VEST — $m.s^{-1}$
Calculé dans frottement.f.
Vitesse estimée pour la loi de paroi.
Version anglaise : *Velocity estimation for the wall function*.
Correspond à la quantité $\overline{u}_{est,a}$ de l'équation (122).

VITC0 DOUBLE PRECISION VITC0 — $m.s^{-1}$
Lu dans lecparam.f, utilisé dans etat.f, entpar.f, forcparois.f, initial.f et pastemps.f.
Vitesse du son.
Version anglaise : *Speed of sound.*
Correspond à la quantité c_0 de l'équation (129).
- Mot-clef VITESSE DU SON.

VX, VZ DOUBLE PRECISION VX(NPMAX), VZ(NPMAX) — $m.s^{-1}$
Initialisés dans initial.f, calculés dans mouvement.f, utilisés dans tablien.f, interpol.f, ecrit.f, frottement.f, forcext.f, impulsion.f, mouvement.f, continuite.f, entpar.f et sorpar.f.
Composantes des vitesses.
Version anglaise : *Velocity components.*
Correspondent à la variable $\underline{u}_a$ de l'équation (128).

VXAB, VZAB DOUBLE PRECISION VXAB(NPMAX), VZAB(NPMAX) — $m.s^{-1}$
Calculés dans tablien.f, utilisés dans gradvit.f, fluxvisq.f et continuite.f.
Différences de vitesses entre particules.

Version anglaise : *Velocity difference between particles.*
Correspondent au vecteur $\underline{u}_{ab}$ de l'équation (126).

X, Z DOUBLE PRECISION X(NPMAX), X(NPMAX) m
Lus dans lecinit.f ou lecsuit.f, calculés dans mouvement.f, utilisés dans tablien.f, interpol.f, ecrit.f, forcext.f, mouvement.f, entpar et sorpar.f.
Positions des particules.
Version anglaise : *Particle positions.*
Correspondent au vecteur $\underline{r}_a$ de l'équation (128).

XAB, ZAB DOUBLE PRECISION XAB(NPMAX,NLIENMAX), XAB(NPMAX,NLIENMAX) m
Calculés dans tablien.f, utilisés dans kernel.f, forcparois.f, tauxdef.f, kequation.f, et fluvisq.f.
Différences de coordonnées entre particules.
Version anglaise : *Coordinate differences between particles.*
Correspond aux composantes du vecteur $\underline{r}_{ab}$ de l'équation (130).

XMIN, XMAX DOUBLE PRECISION XMIN, XMAX m
Lus dans lecparam.f, utilisés dans tablien.f, interpol.f, frottement.f, forcext.f, mouvement.f et sorpar.f.
Abscisses extrêmes du domaine.
Version anglaise : *Minimum and maximum x of the domain*.
- Mots-clefs ABSCISSE MINIMALE DU DOMAINE D'ETUDE et ABSCISSE MAXIMALE DU DOMAINE D'ETUDE.

ZMIN, ZMAX DOUBLE PRECISION ZMIN, ZMAX m
Lus dans lecparam.f, utilisés dans tablien.f, interpol.f, frottement.f, mouvement.f et sorpar.f.
Cotes extrêmes du domaine.
Version anglaise : *Minimum and maximum z of the domain*.
- Mots-clefs COTE MINIMALE DU DOMAINE D'ETUDE et COTE MAXIMALE DU DOMAINE D'ETUDE

2.4 MODE D'EMPLOI DU LOGICIEL

2.4.1 Contexte

2.4.1.1 Rappels sur le domaine de validité et les limites du logiciel

La méthode numérique SPH a déjà fait ses preuves dans le domaine de la mécanique des fluides (voir Monaghan, 1994 ; Issa, 2005). Elle semble posséder la faculté de reproduire des écoulements très divers, et même sortant du champ d'application des méthodes eulériennes ; on peut citer à cet égard le déferlement de la houle, ou encore le phénomène de ressaut hydraulique. Les avantages essentiels de la méthode sont sa simplicité et l'absence de maillage, propriété qui autorise des mouvements

d'ampleur arbitraire et mettant en jeu des distorsions arbitrairement grandes, ainsi que des écoulements non connexes dans l'espace et / ou le temps, et même à la topologie arbitraire et mouvante.

Au vu des cas-tests déjà effectués au LNHE, on peut affirmer que le code SPARTACUS-2D est particulièrement bien adapté à la modélisation d'écoulements à surface libre (voir dossier de validation du code en troisième partie de ce document), à condition de demeurer dans le domaine des écoulements quasi incompressibles. En outre, la possibilité de considérer des écoulements à plusieurs phases peut être largement exploitée pour des applications telles que le mélange de deux fluides non miscibles (voir par exemple Buvat et Violeau, 2006), bien que l'expérience numérique montre qu'il est délicat de simuler des fluides de densités de référence très disparates.

Le code permet aussi d'introduire des parois mobiles. Enfin, son application à la modélisation d'écoulements mixtes (surfaces libres combinées avec des zones à écoulement confiné), comme ceux que l'on peut trouver dans des ouvrages industriels ou maritimes ou des modèles physiques, paraît ne pas poser de difficultés.

Les modèles de turbulence proposés, tels que présentés au **§ 1.1.3**, sont adaptés aux écoulements non-permanent relativement complexes, mais possèdent des limites quant à la précision qu'on est en droit d'attendre des résultats, comme dans tout code de CFD, notamment à cause de la présence éventuelle de surfaces libres complexes, mais aussi de déformations fortes ou d'effets de rotations (voir par exemple Viollet *et al*., 1998).

En tout état de cause, on se souviendra qu'on demeure en deux dimensions, ce qui signifie, conformément à la remarque du bas de la page 11, qu'on considère des écoulements invariants (au sens des grandeurs moyennes turbulentes) selon la direction perpendiculaire au plan de travail.

2.4.1.2 Précautions d'emploi

On veillera, lors de la construction d'un modèle, à vérifier que l'on entre bien dans les limites de validité du code, selon les recommandations du paragraphe précédent (notamment vis-à-vis du caractère bidimensionnel et quasi incompressible de l'écoulement). Il convient également de garder à l'esprit les faits suivants :

- Comme dans tout code de calcul, le choix de la discrétisation joue un rôle déterminant. Il convient de réserver une discrétisation suffisamment raffinée des parties les plus ténues de la géométrie (présence de tubes fins, etc.), tout en limitant le nombre total de particules pour conserver des temps de calcul raisonnables. Cette contrainte est parfois très limitative compte tenu du fait que la taille des particules ne peut varier en espace dans SPARTACUS-2D V1P2. Il est recommandé, avant de construire la géométrie, d'en réaliser un plan soigneux à la main, puis de suivre les recommandations du **§ 2.4.2.1**.
- Il est important d'avoir à l'avance une idée de l'allure de l'écoulement que l'on veut simuler, afin de définir correctement les bornes spatiales du domaine de travail et certains paramètres numériques (notamment la vitesse du son, Cf. **§ 1.2.1.4**).

Afin d'éviter tout malentendu, on aura avantage à se fier aux recommandations du **§ 2.4.2.4** pour le choix des paramètres de calcul, et on consultera avec profit le dossier de validation faisant l'objet de la

3e partie de ce document.

2.4.2 Réalisation d'un calcul

2.4.2.1 Construction de la géométrie

La construction de la géométrie aboutit au fichier d'initialisation (Cf. § **2.1.2.3**). L'utilisateur peut s'aider de la routine geometrie.f, qui ne fait pas partie des sources proprement dites – en cela qu'elle n'est pas appelée par le code – mais est disponible dans /Sources_util/. On y trouve un exemple de programme FORTRAN rédigeant un fichier d'initialisation au format correct (voir un exemple concret en **Annexe 2**). Il est important, lorsque l'utilisateur modifie geometrie.f pour créer la géométrie initiale désirée, de respecter les règles suivantes :

- Les particules peuvent être indexées dans un ordre arbitraire, à condition que les particules fluides (KPAR(I) = 1) soient placées en premier, suivies des particules de bord (KPAR(I) = 2) puis des particules fictives (KPAR(I) = 3) (Cf. **§ 2.1.2.3**).
- Dans le cas où l'utilisateur choisit le modèle des particules fictives (**§ 1.1.4.3**), au-delà de chaque paroi devront être placées plusieurs couches de telles particules. Le nombre de couches doit être la partie entière de h_t / $\delta r = \alpha_t h$ / δr, à laquelle on ajoute 1, c'est-à-dire que ce nombre dépend du facteur h / δr choisi par l'utilisateur (mot-clef `RAPPORT H-DR`). Avec la valeur recommandée de 1,5, et compte tenu des valeurs des supports des noyaux présentés au **§ 1.1.1.1**, le nombre de couches nécessaires est respectivement 3 (noyau d'ordre 3, α_t = 2), 4 (noyau d'ordre 4, α_t = 2,5) et 5 (noyau d'ordre 5, α_t = 3), afin d'assurer aux particules de bord de posséder suffisamment de voisines. Dans les schémas qui suivent, cependant, nous en représenterons systématiquement deux ou trois, par souci de clarté (**figures 15**, **16** et **17**).
- L'utilisateur aura avantage à placer les particules sur une grille carrée de côté δr (espacement initial entre deux particules voisines, comme indiqué au **§ 1.1.1.1**). En particulier, lors de l'établissement de parois courbes, deux possibilités s'offrent *a priori*, décrites sur la **figure 15** :

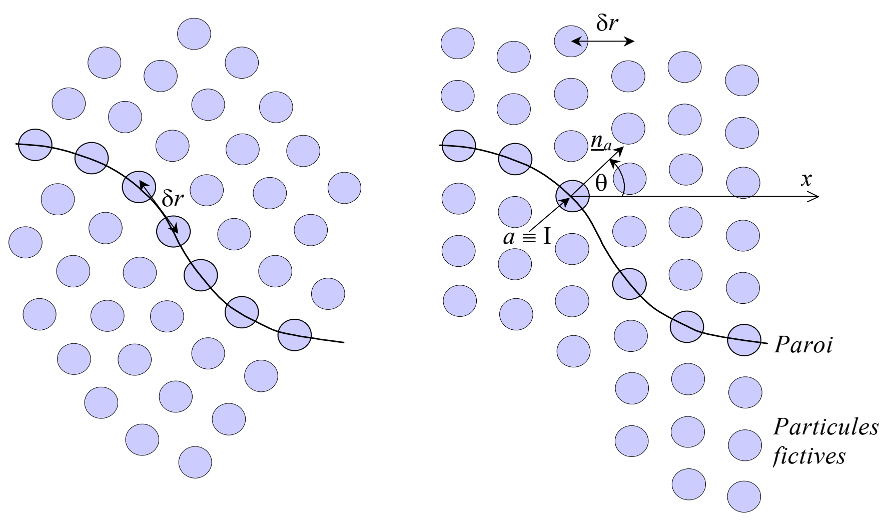


Figure 15 – Exemples de géométrie de paroi courbe : la configuration de droite est recommandée.

Le graphe de gauche montre une paroi le long de laquelle les particules de bord sont espacées de δr (distance curviligne), tandis que celui de droite les montre plus espacées le long de la paroi, tout un présentant un écart δr selon les axes x et z. *De ces deux choix, le second est le bon*. Des tests ont en effet montré que la disposition présentée sur le graphe de gauche aboutit à un mauvais comportement au voisinage immédiat de la paroi. D'autre part, il est clair que la construction de droite est considérablement plus aisée à réaliser. Il peut paraître curieux que la densité linéique de particules de bord y soit variables, ce qui semble créer un « trou » à l'endroit où se trouve le point d'inflexion de la paroi ; la présence de particules fictives à densité surfacique constante vient néanmoins corriger cette lacune.

- Le cas d'un écoulement périodique selon x réclame une précision supplémentaire. Les particules proches des frontières verticales doivent en effet faire l'objet d'un soin particulier. Sur la **figure 16**, les deux frontières verticales coïncident en effet selon le principe indiqué au **§ 1.2.2.1**. Afin que l'espacement virtuel entre les particules des frontières gauche et droite demeure égal à δr, il est indispensable de placer les premières sur la frontière gauche, tandis que les secondes sont situées à une distance δr de la frontière droite (ou inversement).

  Remarque : si l'on travaille dans un canal, on aura intérêt à retenir une configuration périodique. Cependant, la présence éventuelle d'une irrégularité dans la géométrie (corps solide, variation de la cote du fond, etc.) se répercutera sur les caractéristiques de l'écoulement en sortie du canal, et donc à l'entrée par périodicité. Dans ce cas, on prendra donc soin de définir un canal suffisamment long (voir par exemple Buvat et Violeau, 2006).

Enfin, il est bien entendu que chaque particule I doit être affectée de toutes les données requises, dans le bon ordre et au bon format. L'utilisateur a donc intérêt à ne pas modifier les blocs FORTRAN responsables des écritures et des formats dans `geometrie.f`. Ces données sont, écrites dans l'ordre (conformément au **§ 2.1.2.3**) :

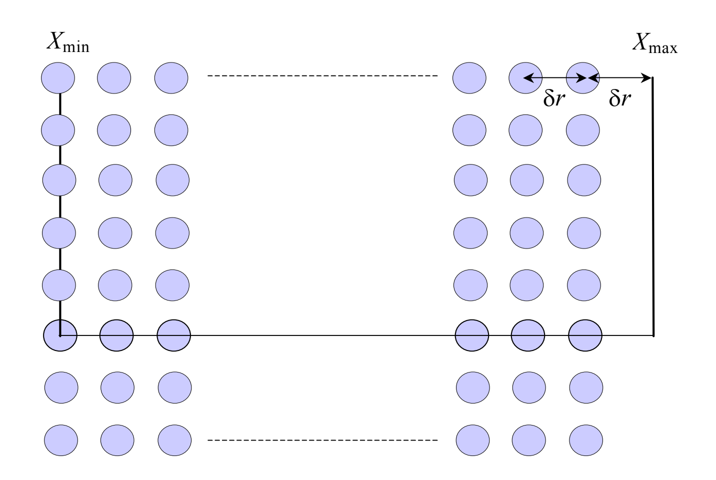


Figure 16 – Disposition initiale des particules dans un cas périodique selon *x*.

- Le numéro de la particule I ;
- Ses coordonnées X(I) et Z(I) ;
- Si I est une particule de bord, l'angle THETA(I–NPARF) compté *en degrés trigonométriques* que fait la normale *intérieure* au bord $\underline{n}_a$ (où $a \equiv$ I) avec l'axe x au point occupé par la particule

de bord I, comme sur la **figure 15** (si I n'est pas une particule de bord, ce nombre n'est pas utilisé et peut donc être spécifié nul). Ce nombre sert dans le code à calculer les tableaux CTHETA(I) et STHETA(I) pour l'établissement des conditions aux limites solides.

- Un label entier KPAR(I) désignant respectivement une particule fluide (KPAR(I) = 1), une particule de bord (KPAR(I) = 2) ou une particule fictive (KPAR(I) = 3). On rappelle qu'elles doivent être listées dans cet ordre ;
- Le nombre KFLUID(I) correspondant au type de fluide de la particule I, pour un écoulement polyphasique (ce nombre peut prendre les valeurs 1, 2, etc. jusqu'à NFLUID, lequel est spécifié dans le fichier des paramètres – Cf. **§ 2.1.2.1** – sous le mot-clef `NOMBRE DE FLUIDES`) ;
- Un entier KENT(I) indiquant si I est une particule dont la vitesse est imposée par l'utilisateur comme indiqué au **§ 2.4.2.3**. Dans ce cas, KENT(I) = 1.
- Dans le cas d'une particule de bord ou fictive, un entier KPARMOB(I) indiquant s'il s'agit d'une paroi mobile (KPARMOB(I) = 1) ou non (KPARMOB(I) = 0). On trouvera plus de détails à ce sujet au **§ 2.4.2.2**.

La routine geometrie.f indique, lors de son exécution, le nombre total de particules, ainsi que les nombres de particules de bord et fictives, ce qui permet à l'utilisateur de spécifier correctement les dimensions des tableaux de travail dans spartacus2d.f (Cf. **§ 2.4.2.4**). Elle vérifie également qu'aucune paire de particules ne coïncide par erreur, et donne enfin les valeurs des limites du domaine créé, ce qui permet à l'utilisateur de spécifier des valeurs idoines pour XMIN, XMAX, ZMIN et ZMAX dans le fichier des paramètres (Cf. **§ 2.4.2.4**). Le domaine peut cependant être étendu par l'utilisateur s'il le souhaite, notamment pour éviter que des particules ne sortent du domaine sous l'effet d'une forte inertie (voir **§ 3.3.2**), mais cela ne peut se faire dans la direction horizontale si l'écoulement est périodique selon *x*.

Enfin, la routine geometrie.f écrit deux fichiers supplémentaires (geometrie.init_global et geometrie.init_bords) qui permettent à l'utilisateur de visualiser respectivement la configuration initiale (toutes particules confondues) et les particules de bord et fictives seules sous Rubens© (voir **§ 2.4.3.1**). La suppression des lignes d'en-tête de geometrie.init_global donne un fichier utilisable sous Tecplot©. Le préfixe « geometrie » de ces fichiers peut être changé sous geometrie.f.

2.4.2.2 Parois mobiles

Si la géométrie envisagée par l'utilisateur comporte des parois mobiles dont le mouvement est spécifié selon une loi connue – il ne s'agit donc pas d'un flotteur dont le mouvement est déterminé par celui du fluide – on procédera comme suit :

- Les particules de bord et fictives constituant la ou les paroi(s) considérées devront être munies, dans le fichier d'initialisation, d'un label KPARMOB(I) égal à 1.
- Dans le fichier des paramètres, le mot-clef `PAROIS MOBILES` doit valoir `OUI` (ou toute autre valeur équivalente, selon le **§ 2.1.2.1**).
- Le mouvement de la ou des paroi(s) doit être spécifié dans la routine utilisateur paroimobile.f (qui doit donc être incluse dans le fichier FORTRAN), sous la forme de deux tableaux donnant les vitesses VXMOB(I) et VZMOB(I) pour chacune des particules (de bord et fictives)

constituant ces parois (il s'agit bien de leur vitesse qui doit être imposée, et non de leur position, conformément à la remarque de la fin du **§ 1.1.4.2**). On rappelle également qu'il est important de ne spécifier que la vitesse normale, suivant la remarque du **§ 1.1.4.4**.

Il pourra être prudent, si le mouvement des parois commence dès le début du calcul, de l'établir progressivement à l'aide d'une « rampe », de manière à éviter de trop brusques discontinuités dans les champs calculés. On veillera enfin à ce qu'au cours de leur trajet, les parois mobiles ne sortent pas du domaine déterminé par les limites spécifiées dans le fichier des paramètres (Cf. **§ 2.4.2.4**). Des exemples sont donnés dans la 3e partie, **§§ 3.2** et **3.3**.

2.4.2.3 Termes moteurs – Flux entrant

Dans certains cas, l'écoulement est mû par la seule gravité ou toute autre force extérieure, ou encore par la présence d'une paroi mobile (voir paragraphe précédent). Dans d'autres, tout particulièrement dans le cas d'un écoulement périodique selon *x*, c'est l'utilisateur qui souhaite imposer un mouvement. Plusieurs possibilités s'offrent alors :

- La première consiste à imposer une vitesse à certaines particules fluides, considérées comme « entrant » dans le domaine. Dans ce cas, le mot-clef `MOTEUR` doit être égal à `1` dans le fichier des paramètres. les particules entrant à l'instant initial doivent être munies, dans le fichier d'initialisation, d'un label KENT(I) égal à 1. Leurs vitesses, pressions, etc. sont spécifiées par l'utilisateur dans la routine `entpar.f`, qui doit alors être incluse dans le fichier FORTRAN. Plus précisément, cette routine permet d'indiquer un critère décidant quand une particule à vitesse imposée devient libre (KENT(I) = 0), auquel cas elle est remplacée par une nouvelle particule dont les paramètres (vitesse, pression, etc.) sont alors imposés par l'utilisateur.

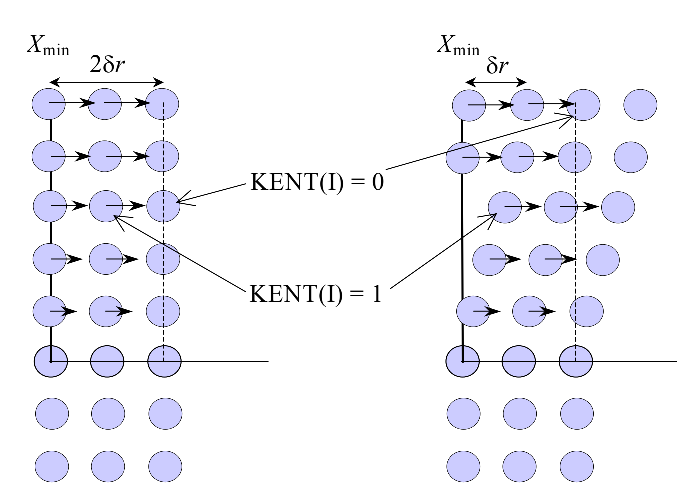


Figure 17 – Exemple de flux de particules à vitesses imposées : elles entrent par la section de gauche et sont progressivement renouvelées au cours de leur mouvement.

Par exemple, dans le cas d'un flux entrant à travers un bord vertical rectiligne situé à gauche, on recommande la disposition de la **figure 17** : deux couches verticales de particules entrantes sont placées initialement. Au cours de leur mouvement, dès que l'une d'elles sort de la zone de largeur $2\delta r$ couvrant ces deux couches, elle devient libre et est aussitôt remplacée par une

autre particule à vitesse imposée située à l'emplacement où elle se trouvait au départ. A l'opposé du domaine, les particules sortent simplement et librement, grâce à la routine sorpar.f (voir **§ 2.2.1.5**). Cette option ne doit pas être retenue si l'écoulement est périodique ; dans ce dernier cas on aura recours aux deux solutions qui suivent.

- Dans le cas d'un écoulement périodique, l'utilisateur peut imposer un débit entrant sans spécifier les vitesses des particules concernées. La première méthode consiste à imposer une force motrice horizontale permettant d'établir ce débit. Dans le cas d'un canal « infini » muni d'une légère pente d'angle α, par exemple, il est préférable de considérer une géométrie rigoureusement horizontale et d'imposer une force horizontale constante égale à $g \sin \alpha \approx g\alpha$. Cette force est alors ajoutée par la routine forcext.f dans les forces extérieures de l'équation de quantité de mouvement (127). Sa valeur F^e correspond au mot-clef `VALEUR DE LA FORCE MOTRICE` du fichier des paramètres, tandis que le mot-clef `MOTEUR` doit valoir `2`.
- Cependant, toujours dans le cas périodique, il est recommandé d'imposer directement la valeur du débit au lieu de la force (pour des raisons de temps de convergence). Le mot-clef `MOTEUR` doit alors valoir `3`, tandis que la vitesse débitante U_{imp} est spécifiée par l'intermédiaire du mot-clef `VALEUR DE LA VITESSE DEBITANTE`, qui correspond au débit divisé par la hauteur de la section à débit imposé. La routine forcext.f calcule alors automatiquement la force correspondante F^e à chaque itération n selon le procédé suivant :

$$F^{e,n} = \frac{2\left(U^n - U_{imp}\right) - \left(U^{n-1} - U_{imp}\right)}{2\delta t} \qquad (156)$$

où U^n représente la vitesse débitante effectivement présente à cet instant à travers la section considérée. Cette relaxation permet le meilleur ajustement de la force motrice, comme cela a été montré dans le cadre du développement d'autres logiciels de type « éléments finis » (voir par exemple Violeau, 2000) ou « volumes finis » (Uribe et Laurence, 2000). La vitesse débitante à l'instant n, notée U_n, est calculée par moyennage des vitesses horizontales des particules se trouvant près de la section gauche du domaine (voir **figure 17**). Cette moyenne prend en compte les particules de bord, sauf en régime laminaire.

On notera que les deux derniers cas présentés exigent que le mot-clef `PERIODICITE SELON X` vaille `OUI`. Des exemples sont donnés dans la 3e partie, **§§ 3.2** et **3.3**.

2.4.2.4 Choix des paramètres de calcul

La rédaction du fichier des paramètres se fait en suivant la nomenclature indiquée au **§ 2.1.2.1**. On en trouvera un exemple en **Annexe 2.1**, ce qui permet de se faire une idée de son allure générale. Le choix des valeurs attribuées aux mots-clefs est bien entendu laissé à l'utilisateur, mais on fera bien de se reporter aux valeurs recommandées au **§ 2.3.2**. Voici en outre quelques règles de bon sens et conformes aux considérations de la 1e partie :

- Il n'est pas recommandé d'imposer une `PERIODE DE SORTIE DES POSITIONS` trop petite, car le fichier des positions est au format ASCII.
- Il convient de choisir des bornes du domaine (`ABSCISSE MINIMALE DU DOMAINE D'ETUDE`, etc.) conforme à ce que donne en sortie le fichier geometrie.f. On pourra

cependant se donner une légère marge de sûreté.

- La vitesse du son doit être estimée selon le procédé décrit au **§ 1.2.1.4**. Lui donner une valeur plus élevée ne peut qu'être bénéfique, sauf en termes de temps CPU.
- Les effets de la turbulence doivent être pris en compte pour des nombres de Reynolds élevés (typiquement supérieurs à 3000), mais ne jouent pas un rôle essentiel pour des écoulements à surface libre dominés par la gravité ou l'inertie (par exemple des vagues en phase de propagation ou une rupture de barrage). En revanche, il est indispensable de les considérer pour reproduire correctement la dynamique d'un écoulement fortement déformé (vague déferlante, voir 3e partie, **§ 3.5**) ou la répartition des vitesses au voisinage d'une paroi (3e partie, **§ 3.1**). Pour tout écoulement où la longueur de mélange est inconnue (c'est-à-dire presque tous les écoulements !) on recommande le modèle k–ε, ou *a minima* le modèle k–L_m avec une longueur de mélange égale à la distance inter-particulaire (par défaut).
- Pour des raisons évidentes de cohérence, la rugosité équivalente ne doit pas excéder (typiquement) la taille d'une particule.
- Un `RAPPORT H-DR` compris entre 1,2 et 1,5 est recommandé.
- L'amortissement (`AMORTISSEMENT SELON X`, etc.) est indispensable pour modéliser une phase de repos (voir **§ 2.4.2.6**).

Non loin après le début du fichier FORTRAN, on trouvera un bloc se présentant comme suit :

```
C Parametres modifiables
C-----------------------
C Adjustable parameters
C----------------------
C
      PARAMETER (NPMAX     = 35000 ,
     .           NBMAX     = 5000  ,
     .           NLIENMAX  = 100   ,
     .           NPARTQMAX = 100   ,
     .           NQUADXMAX = 1040  ,
     .           NQUADZMAX = 160   ,
     .           NPOIN     = 40501 ,
     .           NELEM     = 80000 )
```

Il convient alors de modifier les valeurs indiquées ici en suivant les recommandations suivantes :

- Il est nécessaire d'ajuster le nombre maximal de particules (NPMAX) et le nombre maximal de particules de bord (NBMAX). Ces nombres doivent être compatibles avec les valeurs issues du fichier geometrie.f.
- Le nombre de liens maximum d'une particule NLIENMAX, ainsi que le nombre maximal de particules se trouvant dans un même carré NPARTQMAX peuvent aussi être modifiées, mais les valeurs données ici doivent généralement suffire et peuvent, dans une certaine mesure, être considérées comme universelles.
- Les nombres maximaux de carrés sur les axes x et z (respectivement NQUADXMAX et NQUADZMAX) doivent être ajustés en fonction de la taille du domaine et de la distance entre particules. On rappelle à cet effet (Cf. **§ 1.2.1.1**) que la taille d'un carré est $h_t = \alpha_h \alpha_t \delta r$.

- Dans le cas où l'utilisateur choisirait le post-traitement Rubens©, il est également indispensable de fixer exactement le nombre de points NPOIN et le nombre d'éléments NELEM du maillage retenu pour l'interpolation des champs.

2.4.2.5 Déroulement d'un calcul

Le code se lance grâce à la commande

    spartacus2d [nom du fichier des paramètres]

L'utilisateur voit alors défiler successivement :

- une série d'information concernant les options de lancement, la configuration du système, l'appel des bibliothèques, etc., analogue aux informations fournies par les codes du système TELEMAC d'EDF R&D / LNHE ;
- la liste des fichiers mis en jeu (fichier des paramètres, dictionnaire, fichier d'initialisation, etc.) ;
- un en-tête « SPARTACUS-2D » indiquant le numéro de version ;
- la liste des mots-clefs assortis de leurs valeurs telles qu'elles sont fixées dans le fichier des paramètres ;
- un bloc rappelant les principales informations concernant le nombre de particules.

Voici un exemple (expurgé quant à la liste des mots-clefs) :

```
 SSSSS  PPPPPP   AAAAA  RRRRRR  TTTTTTT  AAAAA   CCCCC  U     U  SSSSS
S     S P     P A     A R     R    T    A     A C     C U     U S     S
S       P     P A     A R     R    T    A     A C       U     U S
 SSSSS  PPPPPP  AAAAAAA RRRRRR     T    AAAAAAA C       U     U  SSSSS
      S P       A     A R     R    T    A     A C       U     U       S
S     S P       A     A R     R    T    A     A C     C U     U S     S
 SSSSS  P       A     A R     R    T    A     A  CCCCC   UUUUU   SSSSS

            22222  DDDDDD        V     V   1 PPPPPP   22222
           2     2 D     D       V     V  11 P     P 2     2
                 2 D     D       V     V 1 1 P     P       2
            22222  D     D       V     V   1 PPPPPP   22222
           2       D     D        V   V    1 P       2
           2       D     D         V V     1 P       2
           2222222 DDDDDD          V       1 P       2222222
_______________________________________________________________________

VALEUR DES MOTS-CLES :

NOMBRE DE PAS DE TEMPS
MOTINT(  9)=    140000

PERIODE DE SORTIE LISTING
MOTINT( 10)=      1400

FIN DU FICHIER POUR DAMOCLES

Nombre de particules
====================
```

```
334299 particules fluides
  3735 particules de bord
 11205 particules fictives
349239 particules
```

Puis, pour le premier pas de temps, ainsi que pour chaque pas de temps multiple de la période de sortie listing (voir le mot-clef correspondant au **§ 2.3**), un bloc donnant :

- le numéro de l'itération ;
- le nombre de carrés utiles à la gestion des liens entre particules (Cf. **§ 1.2.2.1**) ;
- la valeur du pas de temps donné par l'équation (149) ;
- la valeur du temps écoulé (variable TEMPS) ;
- le nombre de particules fluides ;
- si des particules sont sorties et / ou entrées, conformément aux algorithmes décrits aux **§§ 2.2.1.5** et **2.4.2.3**, le nombre de particules sorties et / ou entrées durant l'itération courante, ainsi que les nombres totaux de particules sorties et / ou entrées depuis le début du calcul.

Voici un exemple type de blocs listing conforme à la description qui précède :

```
Iteration :             1
 --------------------------------------------------------
 Nombre de carres :    895 *   103
 Pas de temps     : 0.300E-04 s
 Temps ecoule     : 0.300E-04 s
 Nombre de particules fluides       :        334299

 Iteration :          1400
 --------------------------------------------------------
 Nombre de carres :    895 *   103
 Pas de temps     : 0.300E-04 s
 Temps ecoule     : 0.420E-01 s
 Nombre de particules fluides       :        334299
```

Les éventuels messages d'erreurs ou d'avertissement apparaissent de la même manière, indiquant, par exemple, si un tableau est sous-dimensionné, si deux particules coïncident fortuitement, etc.

2.4.2.6 Phase initiale de repos

Lors de la réalisation d'un calcul, il convient généralement de partir d'une configuration particulaire correspondant à un état physique admissible du système, ce qui est généralement incompatible avec une disposition des particules sur une grille carrée comme il est recommandé au **§ 2.4.2.1**. A cela, une raison simple : les pressions initiales des particules sont alors nulles, puisque leurs volumes, masses et densités sont identiques (voir à ce sujet le **§ 2.2.1.1**). Il est ainsi nécessaire de laisser les particules reposer un certain temps afin que s'établisse un régime de pression hydrostatique. Le processus (automatiquement géré par l'algorithme du code) est le suivant : sous l'effet de la gravité, les particules tendent à descendre légèrement en se tassant très légèrement (d'autant plus qu'on se rapproche du fond), augmentant leur densité *via* l'équation de continuité (126) puis leur pression *via* l'équation d'état (129). A l'issue de ce processus, la densité varie entre le fond et la surface d'environ

1% (voir **§ 1.2.1.4**), si bien que le volume des particules varie d'autant. La distance verticale entre deux particules voisines est donc réduite d'environ 1% (leur distance horizontale est inchangée).

Les éventuelles parois mobiles doivent, dans cette phase de repos, être maintenues fixes, tandis que les forces extérieures imposées (**§ 2.4.2.3**) doivent être nulles, à l'exception de la gravité. Toutefois, l'expérience numérique montre (Issa, 2005) que les particules ont alors tendance à se regrouper par paire, perturbant la résolution spatiale. Aussi est-il recommandé, lors de l'établissement de cette phase, d'avoir recours aux mots-clefs `AMORTISSEMENT SELON X` et `AMORTISSEMENT SELON Z` afin d'introduire dans les projections selon x et z de l'équation de quantité de mouvement des termes d'amortissement prenant les formes respectives

$$\begin{aligned} & -\Gamma_x \overline{u}_{a,x} \\ & -\Gamma_z \overline{u}_{a,z} \end{aligned} \qquad (157)$$

… où Γ_x et Γ_z sont des coefficients constants choisis par l'utilisateur, grâce aux mots-clefs `COEFFICIENT D AMORTISSEMENT SELON X` et `COEFFICIENT D AMORTISSEMENT SELON Z`. Il est recommandé de leur donner une valeur de l'ordre de l'inverse du pas de temps (Issa, 2005). Le nombre de pas de temps correspondant à une telle phase, indépendamment des échelles spatiales, est de l'ordre de 50 000. Ces valeurs, établies au **§ 3.2.2** dans le cas d'un canal au repos, sont ici généralisées et ne constituent donc que des ordres de grandeur.

Une fois cette phase terminée, l'utilisateur peut procéder à une suite de calcul (**§ 2.4.2.7**) pour lancer une phase dynamique, en supprimant l'amortissement. Bien que ce procédé ne soit pas obligatoire (il est possible de commencer immédiatement une phase dynamique à partir de la configuration initiale), l'expérience numérique nous conduit à la recommander dans la plupart des cas d'écoulements gravitaires. Un exemple est donné dans la 3e partie, au **§ 3.2**. Une alternative consiste à débuter directement par une phase dynamique, à condition d'imposer à l'instant initial un profil de pression hydrostatique.

2.4.2.7 Suite de calcul

Si l'utilisateur souhaite poursuivre un calcul terminé (comme par exemple le cas d'une phase statique telle que présentée au paragraphe précédent), il doit utiliser le mot-clef `SUITE DE CALCUL` et le fixer à `OUI`. Dans ce cas, le nom du fichier d'initialisation doit être le FICHIER SUITE du précédent calcul. Le temps physique sera compté à partir du temps final du calcul précédent (il importe donc de lui avoir donné un nom dans le fichier des paramètre dudit calcul précédent, sans quoi ce fichier suite ne sera pas créé). Ce nouveau calcul pourra faire lui-même l'objet d'une suite, selon les mêmes modalités.

Une suite de calcul peut faire l'objet d'un choix de paramètres différents de ceux du calcul précédent, les précautions nécessaires étant programmées à cet effet (voir notamment le **§ 1.2.1.3**).

2.4.3 Exploitation des résultats

Pour la lecture et l'exploitation des résultats, l'utilisateur a le choix entre deux post-processeurs : Rubens©, développé pour le système TELEMAC du LNHE et distribué avec les logiciels de ce système, et Tecplot© (http://www.tecplot.com/). Ce choix est déterminé par le mot-clef `POST-PROCESSEUR` (qui prend les valeurs respectives `1` et `2`). Quel que soit le choix fait, les variables

accessibles au traitement sont :

- les composantes de la vitesse (VX et VZ) ;
- la densité (RHO) ;
- la pression (P) ;
- le taux de déformation scalaire (S) ;
- l'énergie cinétique turbulente (TKE) ;
- son taux de dissipation (EPS) ;
- la viscosité turbulente (NUT) ;

Ce sont les variables disponibles dans le fichier des résultats (voir **§ 2.1.3.2**). L'utilisateur peut également créer une variable privée dans le tableau PRIV, grâce à la routine `prive.f`. Rappelons que l'utilisateur dispose de deux fichiers résultats, l'un (fichier des résultats, **§ 2.1.3.2**) contenant les champs aux formats binaire (Rubens©) ou ASCII (Tecplot©), l'autre contenant les positions au format ASCII (**§ 2.1.3.1**).

2.4.3.1 Utilisation de Rubens©

L'utilisateur peut se référer au document utilisateur du logiciel (Quiquempoix, 1997) pour connaître les fonctionnalités approfondies de Rubens©. Nous donnons toutefois ici les principales informations requises pour son usage correct dans le contexte de SPARTACUS-2D. Rappelons tout d'abord que

Le lancement du logiciel se fait grâce à la commande

```
Rubens
```

L'usage du fichier des résultats se fait de la manière suivante (**figure 18**) :

- Sélectionner en haut le menu « Projet / Nouveau Projet », puis le nom du fichier et donner un nom de répertoire à créer en sélectionnant le format « SERAPHIN » ;
- choisir à gauche un type de graphique (Lignes Isovaleurs, Profil Temporel, etc.), puis tracer un rectangle avec le bouton de gauche de la souris ;
- dans la fenêtre qui s'affiche alors, choisir une variable, définir un instant *t*, les bornes (bouton « Initialisation » puis « Défaut ») puis les diverses options du graphe. Les instants disponibles correspondent aux multiples de la période de sortie des champs multipliés par δt (voir le mot-clef correspondant au **§ 2.3**).

On peut ainsi visualiser, par exemple, des isolignes de vitesse VX interpolées sur le fichier maillage décrit au **§ 2.1.2.4**. Le graphe peut être déplacé, ou sa taille modifiée à l'aide du bouton de gauche de la souris, et divers menus permettent de copier, superposer, séparer, effacer des graphes (menu « Graphe »). Pour modifier une légende, cliquer sur la légende seule.

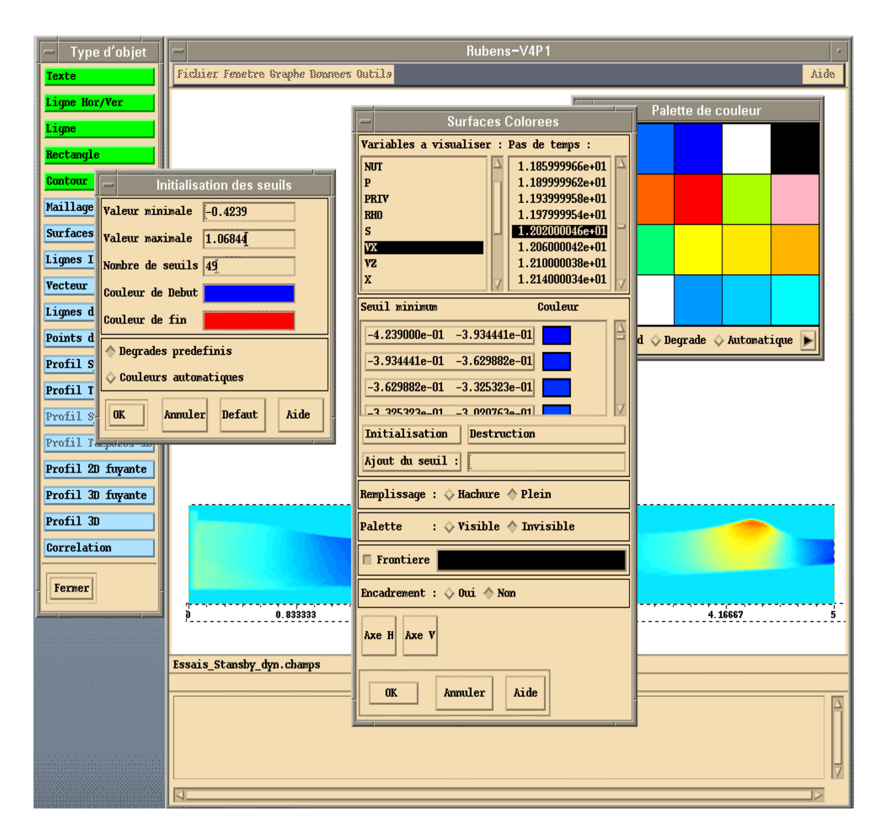


Figure 18 – Visualisation des champs avec Rubens©.

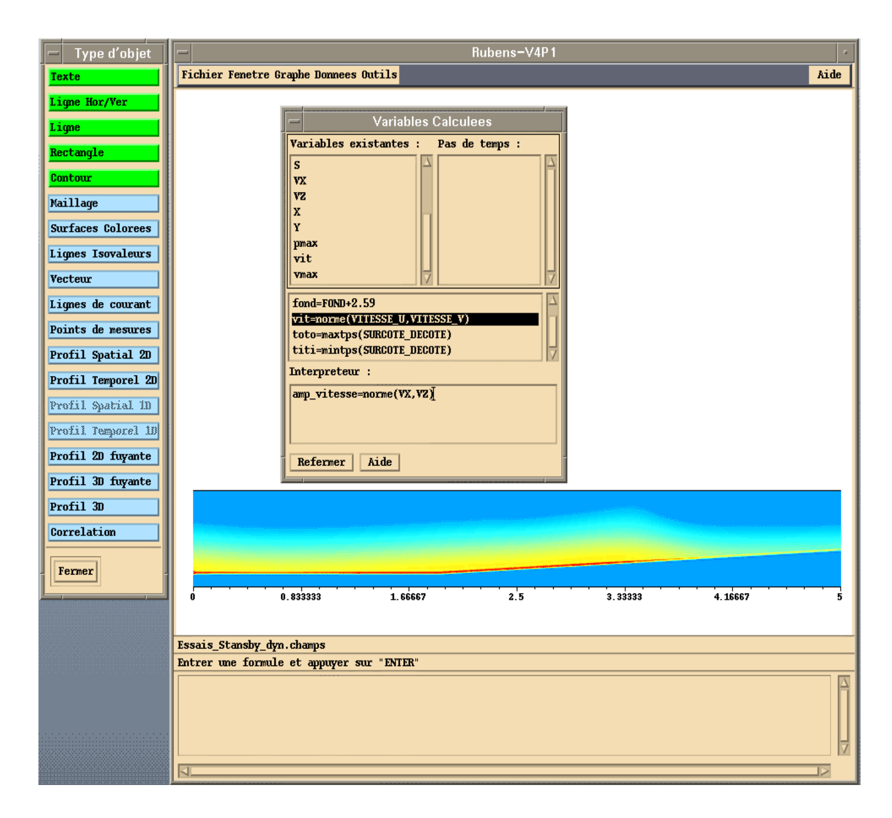


Figure 19 – Définition des variables calculées sous Rubens©.

L'utilisateur peut également créer des variables auxiliaires, en utilisant le menu « Données / Variables Calculées » (**figure 19**). En voici trois exemples :

vect_vitesse = vecteur(VX,VZ)

amp_vitesse = norme(VX,VZ)

tau = S*TKE/EPS

…définissant respectivement un vecteur vitesse, une vitesse scalaire et un temps caractéristique adimensionnel pour les grands tourbillons turbulents. Elles peuvent ensuite être traitées comme les autres ; un vecteur s'affiche grâce au type de graphique « Vecteur » dans la fenêtre de gauche.

On notera enfin que le maillage du fichier correspondant peut être visualisé pour vérification. Toutes les grandeurs étant interpolées aux éléments finis sur ce maillage, les champs visibles sont nécessairement altérés lors de la visualisation.

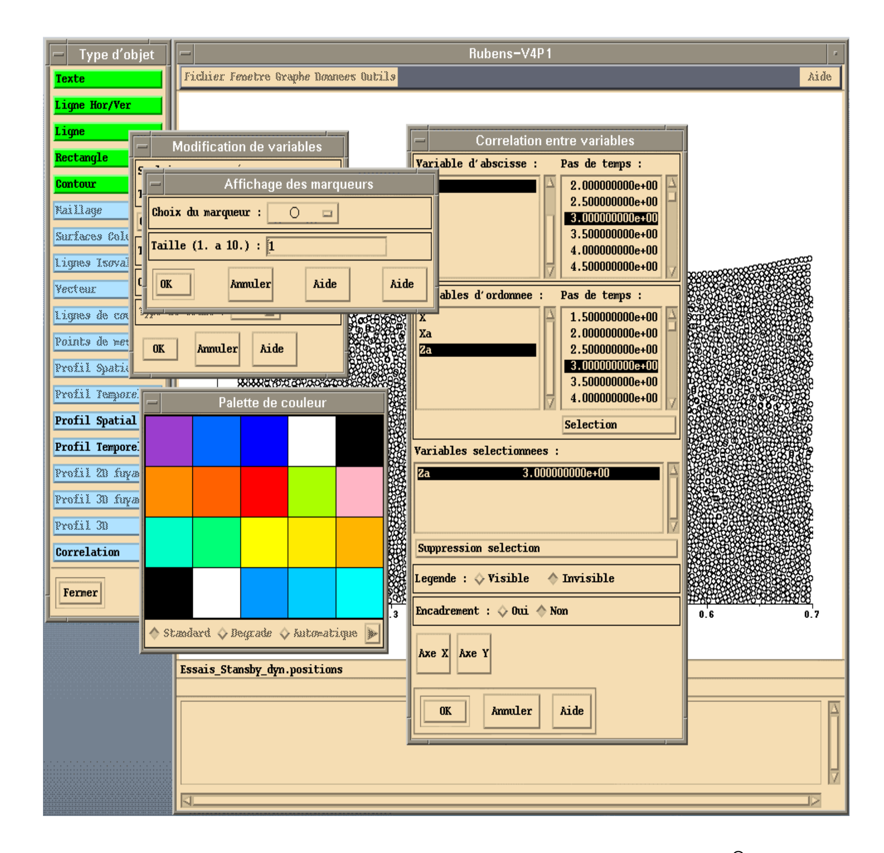


Figure 20 – Visualisation des particules avec Rubens©.

Le fichier des positions permet de visualiser les positions des particules (**figure 20**) :

- il s'ouvre de la même manière, avec le format « SCOPGENE » ;
- choisir un graphe de type « Corrélation », puis sélectionner respectivement « Xa » et « Za » (X et Z) comme abscisse et ordonnée, ainsi qu'un instant t pour chacune des deux (choisir le même instant, bien entendu). Ces derniers correspondent aux multiples de la période de sortie des positions multipliés par δt (voir le mot-clef correspondant au **§ 2.3**) ;

- cliquer sur « Sélectionner ». Il est recommandé de cliquer deux fois sur la sélection apparaissant en-dessous, afin de formater les symboles d'affichage matérialisant les particules (un cercle creux est commode).

Il est particulièrement opportun, dans un contexte lagrangien, de construire une animation permettant de contrôler la dynamique d'un écoulement. On aura alors recours au menu « Outils / Traitement Batch » afin de produire des images successives (**figure 21**). On sélectionnera le projet et l'image concernés, puis le format « Gif_Rubens4 », puis l'option « Maintenant », avant de quitter l'application Rubens©. Enfin, on réalisera l'animation proprement dite à l'aide d'un outil auxiliaire (Rubens© ne le permettant pas dans sa version 4.1).

On rappelle que, lors de la création de la géométrie (**§ 2.4.2.1**), l'utilisateur de Rubens© peut contrôler la disposition des particules dans un fichier nommé geometrie.init_global et celles des particules de bord et fictives dans geometrie.init_bords. Tous deux ouvrables au format « SCOP2D ».

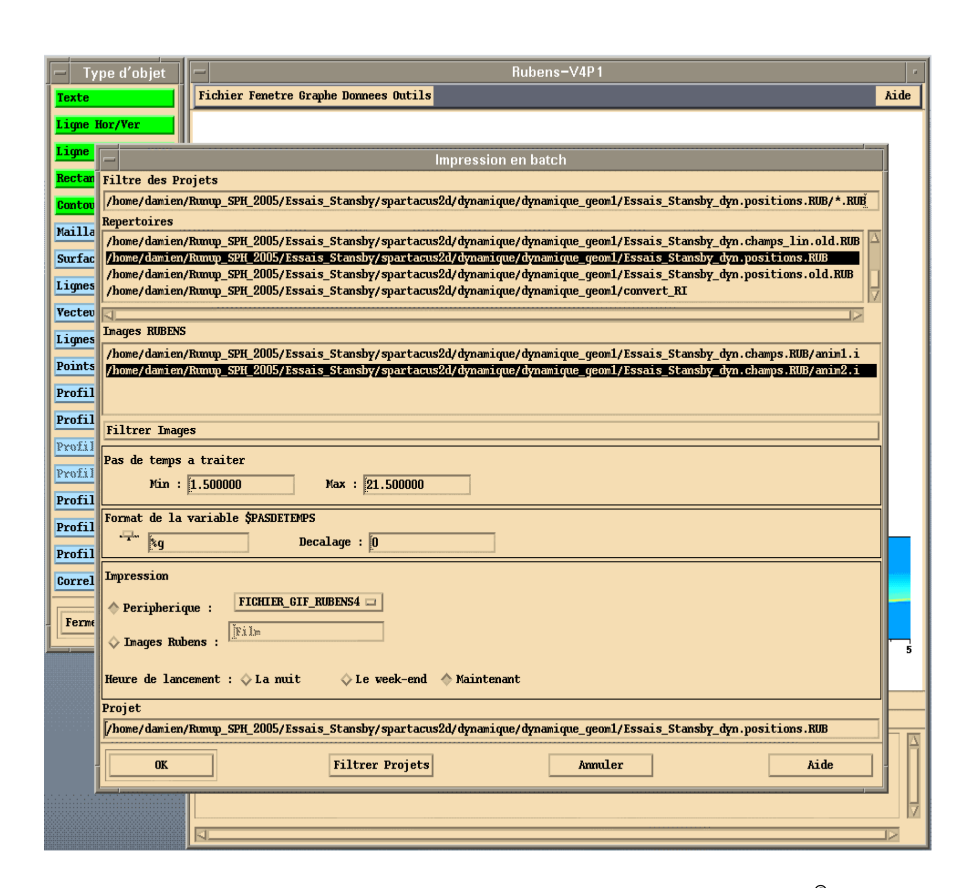


Figure 21 – Lancement d'un traitement batch avec Rubens©.

2.4.3.2 *Utilisation de Tecplot©*

La notice d'utilisation de Tecplot© est livrée à l'acquisition du logiciel. Toutefois, nous allons donner quelques consignes simples permettant de traiter les résultats de SPARTACUS-2D. Il faut d'abord remarquer que l'utilité d'avoir deux fichiers sous Tecplot© est liée à leurs tailles respectives. Contrairement au cas de Rubens©, ici ces deux fichiers ont en effet le même format ; le fichier des résultats est cependant plus lourd car contenant plus de données. Il est donc recommandé de travailler

avec le fichier des positions lorsque cela est suffisant ; tout ce qui suit est néanmoins valable pour le traitement des deux fichiers.

Le chargement d'un fichier peut se faire

- directement au moment du lancement de Tecplot©, par l'instruction

  tecplot [nom du fichier]

- indirectement, une fois Tecplot© prêt à être utilisé, à travers le menu « File » et l'instruction « Load Data File ».

Une fois le fichier résultats chargé, l'utilisateur doit effectuer les opérations suivantes :

- Se placer en visualisation initiale « 2D Cartesian » dans la fenêtre « Select Initial Plot » (**figure 22**). Afin de limiter le temps de traitement, l'utilisateur est invité à sélectionner l'option « Show First Zone Only » dans la fenêtre « Select Initial Plot » : seules les données relatives au premier pas de temps de sortie (correspondant à la 1e zone du fichier résultats) seront alors initialement affichées.

- Désélectionner les modes de visualisation relatifs aux limites (« Boundary ») et au maillage (« Mesh ») et sélectionner celui relatif aux particules (« Scatter ») (**figure 23**).

- Choisir les axes correspondant aux abscisses et aux ordonnées à travers le menu « Plot » et l'instruction « Assign XY » (**figure 24**).

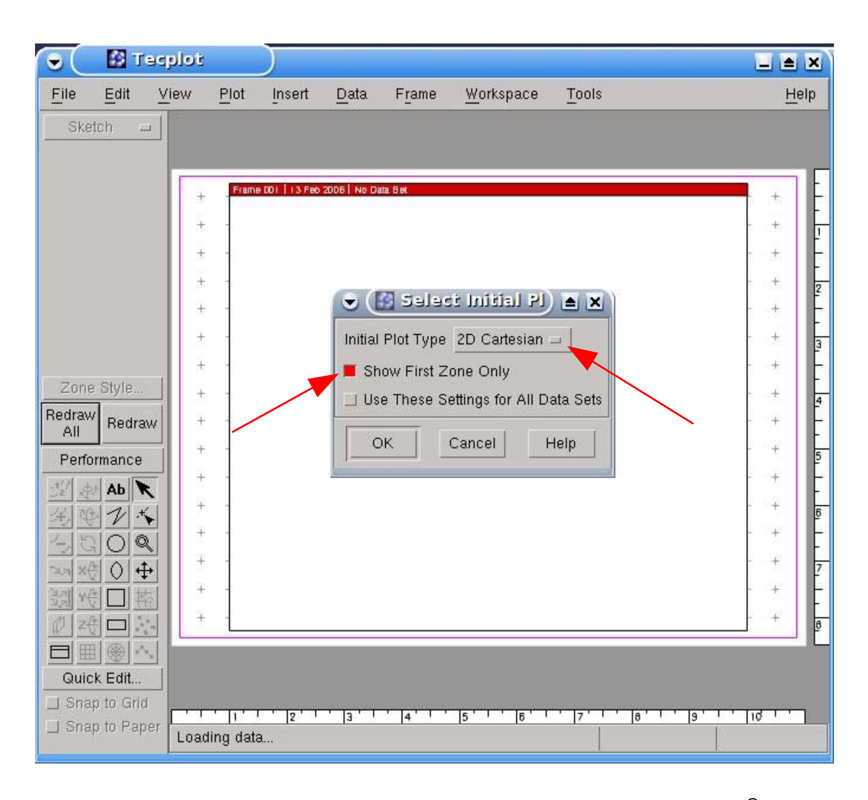


Figure 22 – Ouverture d'un fichier résultat avec Tecplot©.

- La personnalisation des caractéristiques graphiques des particules se fait à travers le menu

« Plot » puis « Zone Style ». Afin d'appliquer les mêmes caractéristiques pour tous les pas de temps de sortie, l'utilisateur est invité à sélectionner toutes les zones du fichier selon « Zone Num » puis « Select All ». La forme des particules est modifiable par l'option « Symbol Shape », les couleurs internes et externes par « Fill Color » et « Outline Color », etc.

Afin de colorer les particules affichées précédemment selon une couleur représentative de l'évolution d'un champs spécifique (pression, vitesse, etc.), l'utilisateur doit effectuer les opérations suivantes :

- Sélectionner le mode de visualisation « Contour » et choisir la variable à visualiser dans la fenêtre « Contour Details » (**figure 25**).
- Modifier la couleur des particules comme indiqué précédemment et choisir « Multi C1 ».
- Les couleurs de la palette utilisée sont modifiables à travers le menu « Plot » puis « Contour Details » et « More ». Après avoir sélectionné l'onglet « Levels », le nombre de niveaux ainsi que les valeurs minimales et maximales sont définis à travers le menu « New Levels ».
- L'affichage de la palette s'effectue à travers le menu « Plot » puis « Contour Details ». Après avoir sélectionné l'onglet « Legend », l'utilisateur peut entièrement gérer l'ensemble des paramètres relatifs à l'affichage de la palette.

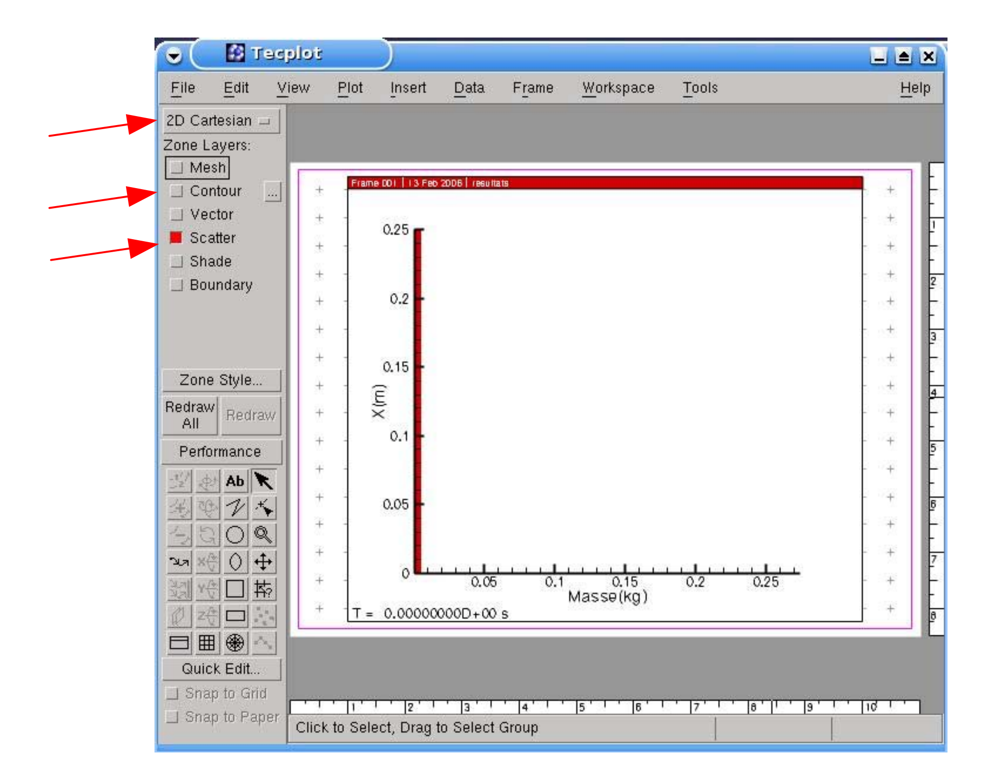


Figure 23 – Sélection des modes de visualisation sous Tecplot©.

Afin de visualiser une répartition de vecteurs vitesse, l'utilisateur doit effectuer les opérations suivantes :

- Sélectionner le mode de visualisation « Vector » et choisir les variables de définition des vecteurs (**figure 26**).

- Les caractéristiques graphiques relatives aux vecteurs sont modifiables à travers le menu « Plot », « Zone Style » puis en sélectionnant l'onglet « Vector ». Encore une fois, il est recommandé à l'utilisateur d'appliquer les mêmes caractéristiques pour tous les pas de temps de sortie.

La visualisation d'un autre pas de temps de sortie s'effectue à travers le menu « Plot » puis « Zone Style ». Sélectionner alors la zone correspondant au pas de temps de sortie souhaité et l'afficher en cliquant sur « Zone Show » puis « Show Selected Only ».

Afin d'effectuer une animation, il est fortement conseillé à l'utilisateur d'appliquer au préalable les mêmes caractéristiques graphiques relatives aux particules pour tous les pas de temps de sortie, ainsi que mentionné précédemment. Il suffit alors, au sein du menu « Tools », de sélectionner « Animate », « Zones », et de préciser les premier et dernier pas de temps de sortie souhaité. Enfin, l'exportation d'images se fait à travers le menu « File », « Export », où l'utilisateur peut alors choisir le format de sortie souhaité.

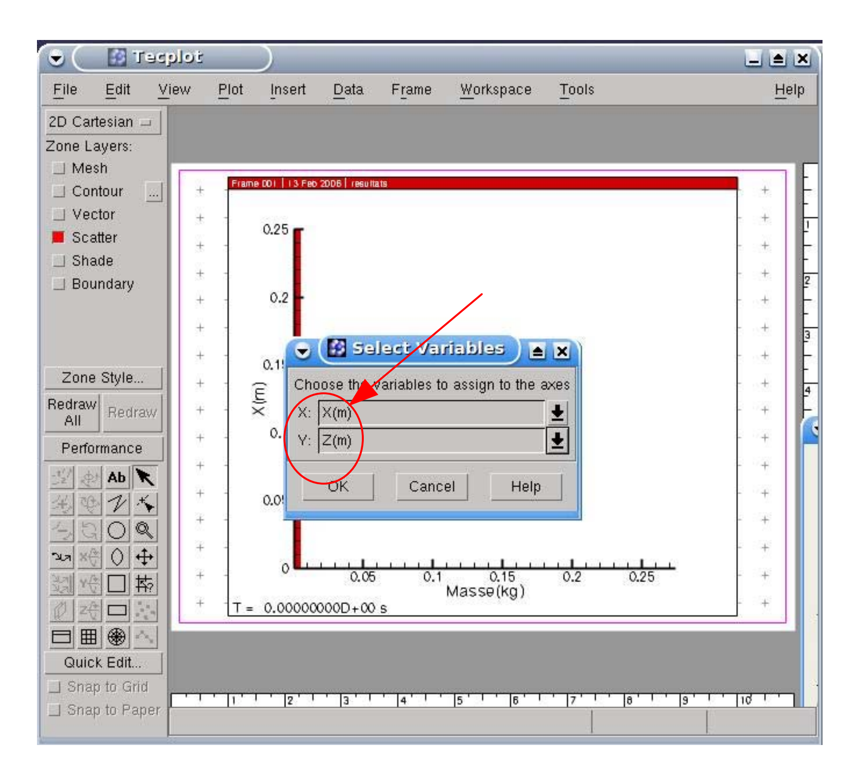


Figure 24 – Choix des axes sous Tecplot©.

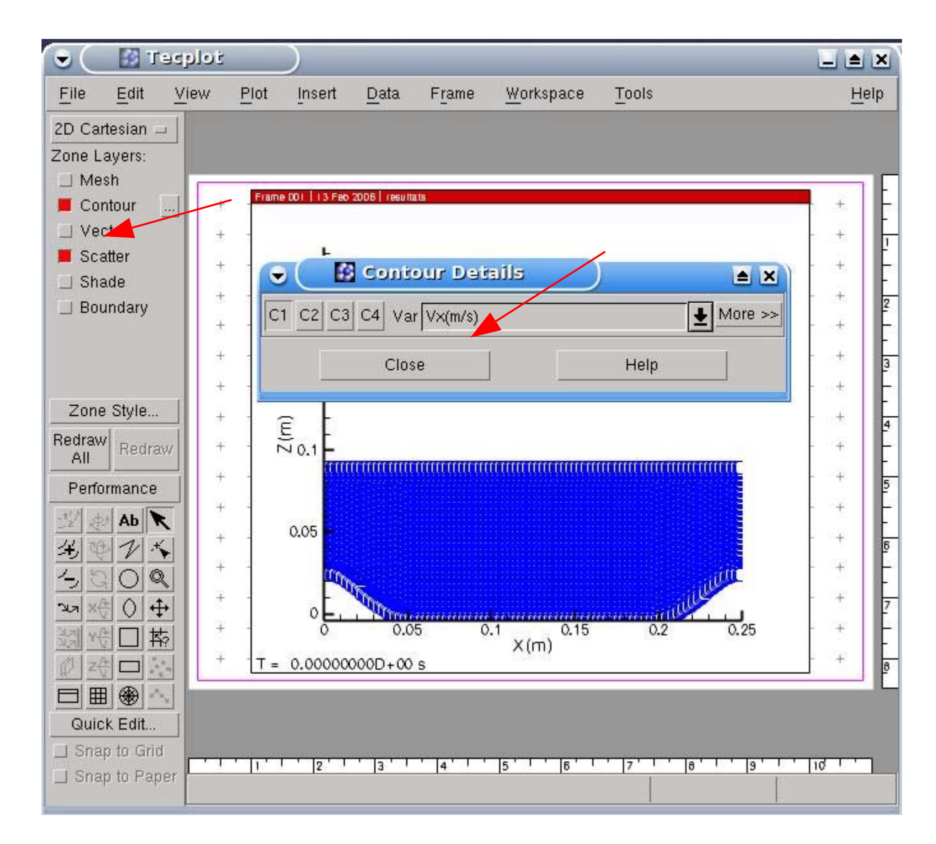


Figure 25 – Sélection du mode « Contour » sous Tecplot©.

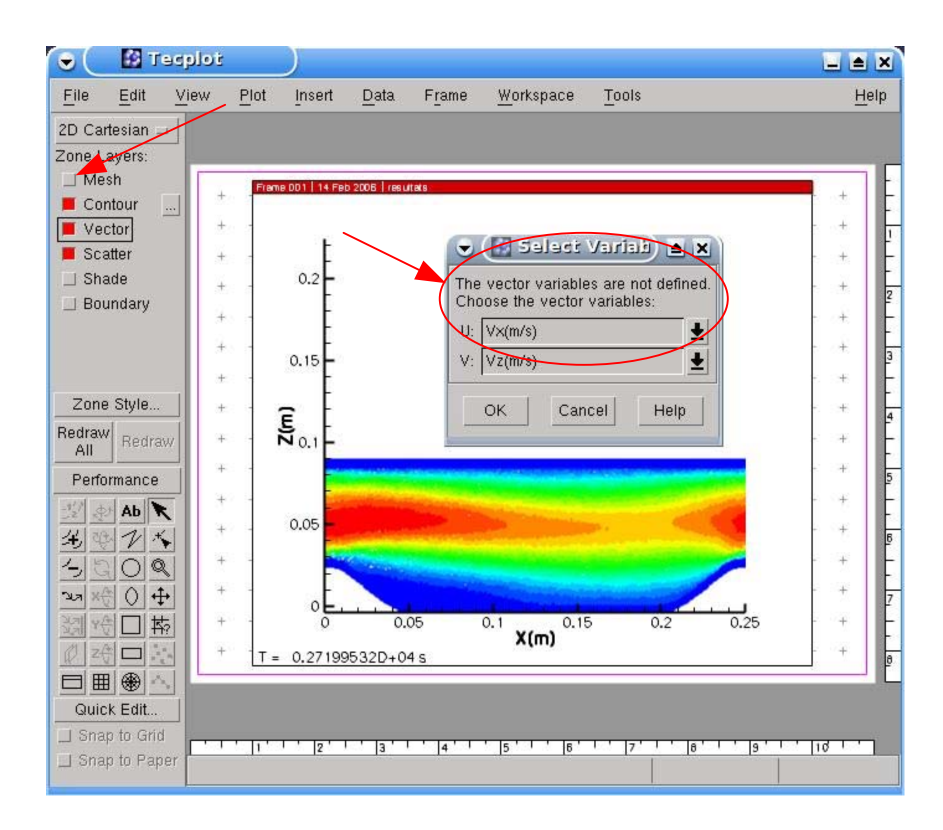


Figure 26 – Visualisation de vecteurs sous Tecplot©.

ANNEXE 2.1 : EXEMPLES DE FICHIERS AUXILIAIRES

Exemple de fichier géométrie

```
C                     *******************
                       PROGRAM GEOMETRIE
C                     *******************
C----------------------------------------------------------------------
C D. Violeau
C +33(0)1-30-87-78-31
C LNHE - 2006
C
C FONCTION : Definit la geometrie initiale du systeme
C FUNCTION : Defines the initial system geometry
C
C PROGRAMMES APPELANT : -
C CALLED BY
C
C PROGRAMMES APPELES  : -
C CALLED PROGRAMS
C
C---------------------------------------------------------------------
C                         MAIN VARIABLES
C .________________.___.______________________________________________
C !            !      !                                               !
C !   NAME     !UNITY !                MEANING                        !
C !____________!______!_______________________________________________!
C !            !      !                                               !
C ! DR         !  M   ! INITIAL INTERPARTICLE SPACING                 !
C ! KPAR       !  -   ! PARTICLE  TYPE                                !
C ! KPER       !  -   ! LOGICAL INDEX FOR PERIODICITY                 !
C ! KFLUID     !  -   ! FLUID TYPE                                    !
C ! KPARMOB    !  -   ! MOVING WALL OR EDGE PARTICLE TYPE             !
C ! NPMAX      !  -   ! MAXIMUM PARTICLE NUMBER                       !
C ! NPARB      !  -   ! NUMBER OF EDGE PARTICLES                      !
C ! NPARF      !  -   ! NUMBER OF FLUID PARTICLES                     !
C ! NPART      !  -   ! TOTAL PARTICLE NUMBER                         !
C ! THETA      !  o   ! ANGLE BETWEEN WALL NORMAL VECTOR AND X-AXIS   !
C ! X, Z       !  M   ! PARTICLE POSITION                             !
C !____________!______!_______________________________________________!
C
C Variables
C==========
C
      IMPLICIT NONE
C
      INTEGER NPMAX , NPARF , NPARB, NPART
      INTEGER I     , K     , J
C
      PARAMETER (NPMAX=30000)
C
      DOUBLE PRECISION DR  , XMIN , XMAX
      DOUBLE PRECISION ZMIN, ZMAX
C
      LOGICAL FLAG
C
      INTEGER KPAR   (NPMAX), KENT(NPMAX), KFLUID (NPMAX)
      INTEGER KPARMOB(NPMAX)
C
```

```
      DOUBLE PRECISION X(NPMAX), Z(NPMAX), THETA(NPMAX)
C
C Placement des particules
C=========================
C Particles placing
C==================
C
      DR=1.D0/100.D0
C
C Particules fluides
C-------------------
C Fluid particles
C----------------
C
      K=0
      DO 103 I=1,100
        DO 102 J=1,200
          K=K+1
          X      (K)=DBLE(I)*DR
          Z      (K)=DBLE(J)*DR
          THETA  (K)=0.D0
          KPAR   (K)=1
          KFLUID (K)=1
          KENT   (K)=0
          KPARMOB(K)=0
 102    CONTINUE
 103  CONTINUE
C
      NPARF=K
C
C Particules de bord
C-------------------
C Edge particles
C---------------
C
C Paroi de gauche
C................
C Left wall
C..........
C
      DO 202 I=1,300
        K=K+1
        X      (K)=0.D0
        Z      (K)=DBLE(I)*DR
        THETA  (K)=0.D0
        KPAR   (K)=2
        KFLUID (K)=1
        KENT   (K)=0
        KPARMOB(K)=0
 202  CONTINUE
C
C Paroi inferieure
C.................
C Bottom wall
C............
C
      DO 204 I=0,400
        K=K+1
        X      (K)=DBLE(I)*DR
        Z      (K)=0.D0
        THETA  (K)=90.D0
```

```
        KPAR   (K)=2
        KFLUID (K)=1
        KENT   (K)=0
        KPARMOB(K)=0
 204  CONTINUE
C
C Paroi de droite
C................
C Right wall
C...........
C
      DO 205 I=1,300
        K=K+1
        X      (K)=4.D0
        Z      (K)=DBLE(I)*DR
        THETA  (K)=180.D0
        KPAR   (K)=2
        KFLUID (K)=1
        KENT   (K)=0
        KPARMOB(K)=0
 205  CONTINUE
C
      NPARB=K-NPARF
C
C Particules fictives
C--------------------
C Mirror particles
C-----------------
C
C Paroi de gauche
C................
C Left wall
C..........
C
      DO 303 I=1,4
        DO 304 J=0,300
          K=K+1
          X      (K)=(DBLE(I)-5.D0)*DR
          Z      (K)= DBLE(J)      *DR
          THETA  (K)=0.D0
          KPAR   (K)=3
          KFLUID (K)=1
          KENT   (K)=0
          KPARMOB(K)=0
 304    CONTINUE
 303  CONTINUE
C
C Paroi inferieure
C.................
C Bottom wall
C............
C
      DO 305 I=1,409
        DO 306 J=1,4
          K=K+1
          X      (K)=(DBLE(I)-5.D0)*DR
          Z      (K)=(DBLE(J)-5.D0)*DR
          THETA  (K)=0.D0
          KPAR   (K)=3
          KFLUID (K)=1
          KENT   (K)=0
```

```
          KPARMOB(K)=0
 306    CONTINUE
 305  CONTINUE
C
C Paroi de droite
C................
C Right wall
C...........
C
      DO 307 I=1,4
        DO 308 J=0,300
          K=K+1
          X       (K)=(DBLE(I)+400.D0)*DR
          Z       (K)= DBLE(J)        *DR
          THETA   (K)=0.D0
          KPAR    (K)=3
          KFLUID  (K)=1
          KENT    (K)=0
          KPARMOB(K)=0
 308    CONTINUE
 307  CONTINUE
C
      NPART = K
C
      PRINT*,''
      PRINT*,NPARF             ,' particules fluides'
      PRINT*,NPARB             ,' particules de bord'
      PRINT*,NPART-NPARF-NPARB,' particules fictives'
      PRINT*,NPART             ,' particules'
      PRINT*,''
C
C Verification de non-coincidence des particules
C-----------------------------------------------
C Checking for particle non-coincidence
C--------------------------------------
C
      FLAG=.TRUE.
      DO 150 I=1,NPART-1
        DO 151 J=I+1,NPART
          IF (ABS(X(I)-X(J)).LT.DR/10.D0.AND.
     .        ABS(Z(I)-Z(J)).LT.DR/10.D0) THEN
            PRINT*,'Attention : particules trop proches en ',X(I),Z(I)
            PRINT*,'             numeros ',I,' et ',J
            FLAG=.FALSE.
          ENDIF
 151    CONTINUE
 150  CONTINUE
      IF (FLAG) PRINT*,'Distances OK'
      PRINT*,''
C
C Recherche des limites
C----------------------
C Search for boundaries
C----------------------
C
      XMIN =  1.D16
      XMAX = -1.D16
      ZMIN =  1.D16
      ZMAX = -1.D16
      DO 160 I=1,NPART
        IF (X(I).LT.XMIN) XMIN=X(I)
```

```
        IF (X(I).GT.XMAX) XMAX=X(I)
        IF (Z(I).LT.ZMIN) ZMIN=Z(I)
        IF (Z(I).GT.ZMAX) ZMAX=Z(I)
 160  CONTINUE
      PRINT*,'XMIN = ',XMIN
      PRINT*,'XMAX = ',XMAX
      PRINT*,'ZMIN = ',ZMIN
      PRINT*,'ZMAX = ',ZMAX
      PRINT*,''
C
C Ecriture du fichier d'initialisation
C-------------------------------------
C Initializing file writing
C-------------------------
C
      OPEN (69,FILE='colonne_eau.init')
C
      WRITE (69,901) NPARF
      WRITE (69,901) NPARB
      WRITE (69,901) NPART
      WRITE (69,902) DR
C
      DO 104 I=1,NPART
        WRITE (69,900) I, X      (I), Z     (I), THETA(I),
     .                    KPAR   (I), KFLUID(I), KENT (I),
     .                    KPARMOB(I)
 104  CONTINUE
C
      CLOSE (69)
C
C Fichiers de visualisation
C.........................
C Visualizing files
C.................

      OPEN (68,FILE='colonne_eau.init_global')
C
      WRITE (68,956) 'D.Violeau'
      WRITE (68,956) '2006'
      WRITE (68,956) 'colonne_eau '
      WRITE (68,955) 'X'
      WRITE (68,955) 'Z'
C
      DO 961 I=1,NPART
        WRITE (68,957) X(I), Z(I)
 961  CONTINUE
C
      CLOSE (68)

      OPEN (67,FILE='colonne_eau.init_bords')
C
      WRITE (67,956) 'D.Violeau'
      WRITE (67,956) '2006'
      WRITE (67,956) 'colonne_eau'
      WRITE (67,955) 'X'
      WRITE (67,955) 'Z'
C
      DO 962 I=NPARF+1,NPART
        WRITE (67,957) X(I), Z(I)
 962  CONTINUE
C
```

```
      CLOSE (67)
C
 955  FORMAT ('''',A9,'''')
 956  FORMAT (A)
 957  FORMAT (2F12.4)
 900  FORMAT (I6,3(D16.8),4(I1))
 901  FORMAT (I6)
 902  FORMAT (D16.8)
C
      END
```

Exemple de fichier d'initialisation

(Les points entre les sections indiquent des séries coupées par souci de concision).

```
18918
983
23853
0.10000000D-01
    1  0.10000000D-01 0.10000000D-01 0.00000000D+00    1    1    0    0
    2  0.10000000D-01 0.20000000D-01 0.00000000D+00    1    1    0    0
    3  0.10000000D-01 0.30000000D-01 0.00000000D+00    1    1    0    0
    4  0.10000000D-01 0.40000000D-01 0.00000000D+00    1    1    0    0
    5  0.10000000D-01 0.50000000D-01 0.00000000D+00    1    1    0    0
      .                 .               .                .    .    .    .
      .                 .               .                .    .    .    .
      .                 .               .                .    .    .    .
18917  0.76200000D+01 0.35000000D+00 0.00000000D+00    1    1    0    0
18918  0.76300000D+01 0.35500000D+00 0.00000000D+00    1    1    0    0
18919  0.00000000D+00 0.10000000D-01 0.00000000D+00    2    1    0    1
18920  0.00000000D+00 0.20000000D-01 0.00000000D+00    2    1    0    1
      .                 .               .                .    .    .    .
      .                 .               .                .    .    .    .
      .                 .               .                .    .    .    .
19900  0.86000000D+01 0.49000000D+00 0.18000000D+03    2    1    0    0
19901  0.86000000D+01 0.50000000D+00 0.18000000D+03    2    1    0    0
19902 -0.40000000D-01 0.10000000D-01 0.00000000D+00    3    1    0    1
19903 -0.40000000D-01 0.20000000D-01 0.00000000D+00    3    1    0    1
      .                 .               .                .    .    .    .
      .                 .               .                .    .    .    .
      .                 .               .                .    .    .    .
23849  0.86400000D+01 0.46000000D+00 0.00000000D+00    3    1    0    0
23850  0.86400000D+01 0.47000000D+00 0.00000000D+00    3    1    0    0
23851  0.86400000D+01 0.48000000D+00 0.00000000D+00    3    1    0    0
23852  0.86400000D+01 0.49000000D+00 0.00000000D+00    3    1    0    0
23853  0.86400000D+01 0.50000000D+00 0.00000000D+00    3    1    0    0
```

Exemple de fichier des paramètres (en Français)

```
/ Fichiers entrees-sorties
/-------------------------
FICHIER FORTRAN = './colline_lam.f'
FICHIER INITIALISATION = '../GEOMETRIE/colline_lam.init'
FICHIER SUITE =  '../RESULTATS/colline_lam_linux.suite'
FICHIER DES POSITIONS =  '../RESULTATS/colline_lam_position_linux.dat'
FICHIER DES RESULTATS = '../RESULTATS/colline_lam_champs_linux.dat'

/ Parametres de calcul
/---------------------
NOMBRE DE PAS DE TEMPS = 170000
SUITE DE CALCUL =  NON
PERIODE DE SORTIE LISTING =  10000
PERIODE DE SORTIE DES CHAMPS = 10000
PERIODE DE SORTIE DES POSITIONS = 170000

/ Parametres geometriques
/------------------------
ABSCISSE MINIMALE DU DOMAINE D ETUDE  =  0.0
ABSCISSE MAXIMALE DU DOMAINE D ETUDE  = 0.251
COTE MINIMALE DU DOMAINE D ETUDE  =  -4.E-3
COTE MAXIMALE DU DOMAINE D ETUDE  =  8.898E-2
PERIODICITE SELON X =  OUI

/ Parametres physiques
/---------------------
GRAVITE =  NON
NOMBRE DE FLUIDES =  1
DENSITES DE REFERENCE =  1000.
VITESSE DU SON =  0.03
VISCOSITES MOLECULAIRES = 1.E-6

/ Turbulence
/-----------
MODELE DE TURBULENCE =  0
MODELE DE DEFORMATION = 2
MODELE DE PRODUCTION = 2
MODELE DE VITESSE DE FROTTEMENT = 1
RUGOSITE EQUIVALENTE = 0.001
RAPPORT DELTA-DR =  0.1

/ Modelisation des parois
/------------------------
PAROIS MOBILES = NON
TYPE DE CONDITION DE PAROI =  1
TYPE DE FORCES DE PAROI = 1
COEFFICIENT DES FORCES DE PAROI = 100.

/ Moteur de l'ecoulement selon x
/-------------------------------
MOTEUR =  3
VALEUR DE LA FORCE MOTRICE = 0.
VALEUR DE LA VITESSE DEBITANTE = 1.785E-3

/ Parametres numeriques
/----------------------
MODELE DE GRADIENT DE PRESSION = 2
MODELE DE VISCOSITE = 1
```

```
TYPE DU NOYAU = 2
RAPPORT H-DR = 1.2

/ Artifices numeriques
/----------------------
AMORTISSEMENT SELON X =  NON
COEFFICIENT D AMORTISSEMENT SELON X =  0.
AMORTISSEMENT SELON Z =  NON
COEFFICIENT D AMORTISSEMENT SELON Z =  0.
LISSAGE DE LA DENSITE = NON
FACTEUR DE LISSAGE = 0.

/ Visualisation graphique
/-------------------------
POST-PROCESSEUR =  2

&ETA
&FIN
```

ANNEXE 2.2 : REFERENCES DE LA DEUXIEME PARTIE

[1] Buvat C. et Violeau D. (2006), *Modélisation hydrodynamique de barrages flottants anti-hydrocarbure en présence de houle et de courant à l'aide du logiciel SPARTACUS-2D (projet SIMBAR)*, rapport EDF R&D / LNHE n° H-P75-2006-01487-FR.

[2] Issa R. (2005), *Numerical assessment of the Smoothed Particle Hydrodynamics gridless method for incompressible flows and its extension to turbulent flows*, rapport de Thèse de l'Université de Manchester (Royaume-Uni), Department of Mechanical, Aerospace and Manufacturing Engineering.

[3] Peron S., Prygoda P. (1997), *Mailleur MATISSE, Manuel Utilisateur version 1.0*, rapport EDF R&D / LNH n° HE-45/97/022/A.

[4] Quiquempoix O. (1997), *Rubens Version 4.1. Manuel utilisateur*, rapport EDF R&D / LNH n° HE-45/97/008/B.

[5] Uribe J.C. et Laurence D. (2000). *10th joint Ercoftac (SIG-15)/IAHR/QNET-CFD Workshop on Refined Turbulence Modelling*.

[6] Viollet P.-L., Chabard J.-P, Esposito P. et Laurence D. (1998), *Mécanique des fluides appliquée. Ecoulements incompressibles dans les circuits, canaux, rivières, autour de structures et dans l'environnement*, Presses de l'Ecole Nationale des ponts et Chaussées.

[7] Violeau D. (2000), *MAST-3 COSINUS project. Intercomparison of the preliminary results of several numerical models for a 1DV case*, rapport EDF R&D / LNH n° HP-72/2000/005/A.

3. TROISIEME PARTIE : DOSSIER DE VALIDATION

3.1 GENERALITES

3.1.1 Principe de la validation

Les cas tests présentés ici ont pour objet de valider le code SPARTACUS-2D V1P2 et d'en montrer les fonctionnalités. Ils ont été choisis d'après les simulations typiques présentées dans les publications disponibles, ainsi que dans l'arsenal de cas testés à EDF R&D dans le cadre de l'hydrodynamique à surface libre, et sont classés dans un ordre arbitraire. Nous avons choisi ces cas de sorte que le plus grand nombre possible de phénomènes physiques y soient représentés, et avons à chaque fois retenu les options physiques et numériques les plus appropriées ; dans certains cas toutefois, l'influence de tel ou tel choix de modélisation est examinée, afin de renseigner l'utilisateur sur les avantages et les inconvénients de chacun.

3.1.2 Nomenclature retenue

Chaque cas-test fait l'objet d'un paragraphe présenté comme une fiche auto-portante. On y résume les caractéristiques géométriques et physiques de la simulation, ainsi que les options retenues (mots-clefs). Le fichier des paramètres est listé, puis on présente les résultats sous forme de graphiques commentés, présentant notamment, lorsque cela est possible, des comparaisons avec des résultats théoriques, ou encore avec des séries de mesures ou des simulations de référence provenant de codes antérieurs déjà validés, si possible issues de publications.

3.2 FICHE N°1 : CANAL A SURFACE LIBRE PERIODIQUE

- Rédacteur : D. VIOLEAU & R. ISSA
- Date de rédaction de la fiche : 12 décembre 2006
- Logiciel utilisé : SPARTACUS-2D Version 1.2 (V1P2)
- Pré-processeur utilisé : -
- Post-processeur utilisé : Tecplot© – Version 360
- Machine de calcul : PC Linux – Calibre 4 – 2.8 GHz, 2Go RAM

3.2.1 Description physique

Il s'agit de tester la capacité du code à prédire finement la distribution des vitesses et de certaines grandeurs turbulentes (énergie cinétique, taux de dissipation, viscosité turbulente) à haut nombre de Reynolds. On prend donc l'exemple d'un canal « infini », c'est-à-dire représenté par un écoulement périodique selon *x*, physiquement mû par une légère inclinaison du lit, représentée ici artificiellement par une force motrice horizontale (Cf. **§ 2.4.2.3**). La validité des conditions aux limites turbulentes aux parois sera de la sorte également testée, ainsi que le modèle de périodicité spatiale.

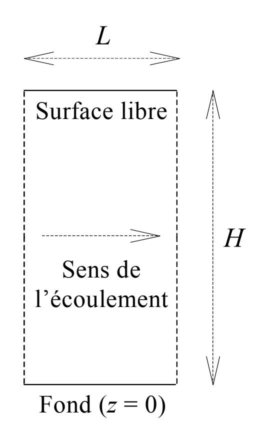


Figure 27 – Géométrie du canal périodique à surface libre.

La **figure 27** montre la géométrie retenue, avec une origine des cotes fixée au fond du canal. La hauteur d'eau H est égale à 1,0 m, tandis que la vitesse moyenne débitante est fixée à U_{imp} = 1,0 m/s et la rugosité du fond à k_s = 1 cm. A noter que le choix de ces grandeurs est arbitraire mais sans effet sur les résultats puisque les validations seront présentées sous forme adimensionnelles (**§ 3.2.3**). Le fluide est de l'eau (ρ_0 = 1000 kg/m^3, $\nu = 10^{-6}$ m^{2}s^{-1}) et le nombre de Reynolds (basé sur la vitesse moyenne débitante et la hauteur d'eau) vaut donc Re = 10^6.

3.2.2 Paramètres numériques

A l'instant initial, les particules sont espacées d'une distance δr = 1 cm sur une grille régulière. Un noyau d'ordre 4 (KKERNEL = 2) est ici considéré pour un rapport h / δr = 1,5. Les parois sont

modélisées par des particules de bord et quatre couches de particules fictives, selon le procédé décrit au **§ 2.4.2.1** (INT(h_t / δr) + 1 = 4). Conformément à la recommandation faite au **§ 2.4.2.1** au sujet des écoulements périodiques, la largeur L est fixée à 0,1 m (supérieure à 2 fois le support du noyau = 2 × h_t = 0,075 m). La discrétisation spatiale du système (fluide et parois solides) est décrite en **table 1**.

Particules fluides	1000
Particules de bord	10
Particules fictives	40
Nombre total de particules	1050

Table 1 – Nombre de particules mises en jeu.

La densité des particules est initialement fixée à la densité de référence et leur pression est nulle, puis on effectue une phase de repos comme présenté au **§ 2.4.2.6**, avant de lancer la phase dynamique comme suite de calcul pour aboutir à un régime permanent. Pour la phase de repos, conformément au **§ 2.4.2.6**, la vitesse du son est fondée sur $\sqrt{gH} = 3{,}13$ m/s. Conformément aux recommandations du **§ 1.2.1.4**, nous prendrons donc c_0 = 32 m/s. Dans ce cas géométrique simple, la solution théorique transitoire des équations du mouvement (26), (32) et (58) s'écrit

$$\frac{\overline{u}_z}{g\tau} = -\frac{4}{\pi}\exp\frac{-t}{\tau}\left(\sum_{n\le n_0}\frac{\tau_n/\tau}{2n+1}\operatorname{sh}\frac{t}{\tau_n}\sin\frac{z}{\zeta_n} + \sum_{n>n_0}\frac{\tau_n/\tau}{2n+1}\sin\frac{t}{\tau_n}\sin\frac{z}{\zeta_n}\right) \tag{158}$$

$$\frac{\overline{p}}{\rho_0 gH} = 1 - \frac{z}{H} - \frac{8}{\pi^2}\exp\frac{-t}{\tau}\left[\begin{aligned}&\sum_{n\le n_0}\frac{\zeta_n/\zeta}{2n+1}\left(\frac{\tau_n}{\tau}\operatorname{sh}\frac{t}{\tau_n} + \operatorname{ch}\frac{t}{\tau_n}\right)\cos\frac{z}{\zeta_n}\\ &+\sum_{n>n_0}\frac{\zeta_n/\zeta}{2n+1}\left(\frac{\tau_n}{\tau}\sin\frac{t}{\tau_n} + \cos\frac{t}{\tau_n}\right)\cos\frac{z}{\zeta_n}\end{aligned}\right] \tag{159}$$

avec les notations suivantes :

$$\zeta_n = \frac{\zeta}{2n+1} \qquad \begin{aligned}\frac{1}{\tau_n} &= \frac{1}{\tau}\sqrt{1-\left(\frac{2n+1}{\theta}\right)^2} \quad \text{si} \quad n \le n_0\\ &= \frac{1}{\tau}\sqrt{\left(\frac{2n+1}{\theta}\right)^2 - 1} \quad \text{sinon}\end{aligned} \tag{160}$$

et

$$\begin{aligned}\zeta &= \frac{2H}{\pi} & \tau &= \frac{2}{\Gamma_z}\\ \theta &= \frac{\zeta}{c_0\tau} & n_0 &= \left[\frac{\theta-1}{2}\right]+1\end{aligned} \tag{161}$$

(le crochet désignant la partie entière). Ainsi, le profil de pression tend bien vers une distribution hydrostatique et les vitesses verticales vers zéro. L'équation (158) montre que ces dernières sont majorées par $g\tau$ / 2, si bien que les forces d'amortissement (157) reçues par les particules sont majorées par g. La première condition minorant le pas de temps dans (149) atteint donc la valeur

spécifiée par la seconde condition (CFL) si

$$c_0 \geq 1{,}6\sqrt{gh} \tag{162}$$

… condition effectivement vérifiée. Le pas de temps de simulation est donc fixé ici par la condition de CFL de l'équation (149), et vaut $1{,}875.10^{-4}$ s. Quant aux coefficients d'amortissement, ils possèdent également une signification purement numérique, et leur rôle montre qu'ils devraient avoir des valeurs de l'ordre de grandeur de $1 / \delta t$. Nous choisirons ici $\Gamma_x = \Gamma_z = 5000\ \text{s}^{-1}$. la résolution approchée des équations (158) à (161) montre en outre que le temps nécessaire pour converger vers un profil de pression hydrostatique est de l'ordre de $5{,}0.10^4\tau$. On doit donc choisir ainsi un nombre d'itérations

$$\mathrm{NT} \approx 100\ 000 \tag{163}$$

(il s'agit d'un ordre de grandeur recommandé pour la simulation de toute phase de repos). Nous retiendrons ici NT = 200 000, afin de s'assurer de la convergence et la stabilité de la solution. Le fichier cas est présenté ci-dessous pour la phase de repos :

```
/ Fichiers entrees-sorties
/-------------------------
FICHIER FORTRAN = './canal_turb_statique.f'
FICHIER INITIALISATION = '../GEOMETRIE/canal.init'
FICHIER SUITE =
'../RESULTATS/canal_turb_statique_v1p2_suite'
/FICHIER DE MAILLAGE = '../maillage3'
FICHIER DES POSITIONS =
'../RESULTATS/canal_turb_statique_v1p2_positions'
FICHIER DES RESULTATS =
'../RESULTATS/canal_turb_statique_v1p2_champs'

/ Parametres de calcul
/---------------------
NOMBRE DE PAS DE TEMPS = 200000
SUITE DE CALCUL =  NON
PERIODE DE SORTIE LISTING = 20000
PERIODE DE SORTIE DES CHAMPS = 2000
PERIODE DE SORTIE DES POSITIONS = 200000

/ Parametres geometriques
/------------------------
ABSCISSE MINIMALE DU DOMAINE D ETUDE = 1.E-2
ABSCISSE MAXIMALE DU DOMAINE D ETUDE = 0.11
COTE MINIMALE DU DOMAINE D ETUDE = -4.E-2
COTE MAXIMALE DU DOMAINE D ETUDE = 1.5
PERIODICITE SELON X =  OUI

/ Parametres physiques
/---------------------
GRAVITE = OUI
NOMBRE DE FLUIDES = 1
DENSITES DE REFERENCE = 1000.
VITESSE DU SON = 32.
VISCOSITES MOLECULAIRES = 0.000001

/ Turbulence
/-----------
MODELE DE TURBULENCE = 0
MODELE DE DEFORMATION = 1
MODELE DE PRODUCTION = 2
```

```
MODELE DE VITESSE DE FROTTEMENT = 2
RUGOSITE EQUIVALENTE = 0.01
RAPPORT DELTA-DR =  0.1
RAPPORT DEST-DR =  5.0

/ Modelisation des parois
/------------------------
PAROIS MOBILES = NON
TYPE DE CONDITION DE PAROI = 1

/ Moteur de l'ecoulement selon x
/-------------------------------
MOTEUR = 0
VALEUR DE LA VITESSE DEBITANTE = 1.0

/ Parametres numeriques
/----------------------
MODELE DE GRADIENT DE PRESSION = 2
MODELE DE VISCOSITE = 1
TYPE DU NOYAU = 2
RAPPORT H-DR = 1.5

/ Artifices numeriques
/---------------------
AMORTISSEMENT SELON X = OUI
/COEFFICIENT D AMORTISSEMENT SELON X = 5000.
AMORTISSEMENT SELON Z = OUI
/COEFFICIENT D AMORTISSEMENT SELON Z = 5000.
LISSAGE DE LA DENSITE = NON
/FACTEUR DE LISSAGE = 0.05

/ Visualisation graphique
/------------------------
POST-PROCESSEUR = 2

&ETA
&FIN
```

La **figure 28** montre que le profil de pression obtenu est très proche du profil hydrostatique, bien que de petites oscillations (sans conséquences pour la suite) subsistent près du fond.

Au début de la phase dynamique, les particules sont animées d'un profil de vitesses linéaire entre le fond et la surface, selon la loi

$$\overline{u}(z,t=0)=u_{bord}+\frac{U_{\max}-u_{bord}}{H-\delta}z-\frac{U_{\max}-u_{bord}}{H-\delta}\delta \tag{164}$$

La vitesse u_{bord} représente la vitesse des particules de bord et est de nouveau estimée par la loi logarithmique précédente prise en $z = \delta$, estimée en considérant le rapport δ / δr = 0,1. La vitesse maximale de l'écoulement est atteinte en surface et donnée par le profil de vitesse théorique, qui prend la forme logarithmique (112) (voir Viollet *et al.*, 1998). Nous avons donc :

$$U_{\max}=\overline{u}_{\log}(z=H)=u_*\left(\frac{1}{\kappa}\ln\frac{H}{k_s}+C_*\right) \tag{165}$$

avec les notations du **§ 1.1.4.4**. La vitesse de frottement u_* (ici constante en espace) est donnée par une intégration verticale de la loi logarithmique (112) :

$$U_{imp} = \frac{1}{H}\int_{k_s}^{H} u_{\log}(z) \approx U_{imp} = u_*\left[\frac{H}{\kappa(H-\delta)}\ln\frac{H}{k_s} - \frac{\delta}{\kappa(H-\delta)}\ln\frac{\delta}{k_s} - \frac{1}{\kappa} + C_*\right] \tag{166}$$

Nous obtenons u_* = 5,77 cm/s et U_{max} = 1,14 m/s. Conformément aux recommandations du **§ 1.2.1.4**, la vitesse du son et le pas de temps sont inchangés par rapport à la phase statique, la condition portant sur la hauteur d'eau dans (151) étant dominante. Le nombre d'itérations choisi NT = 200 000 permet largement d'établir un régime dynamique que l'on peut considérer comme permanent.

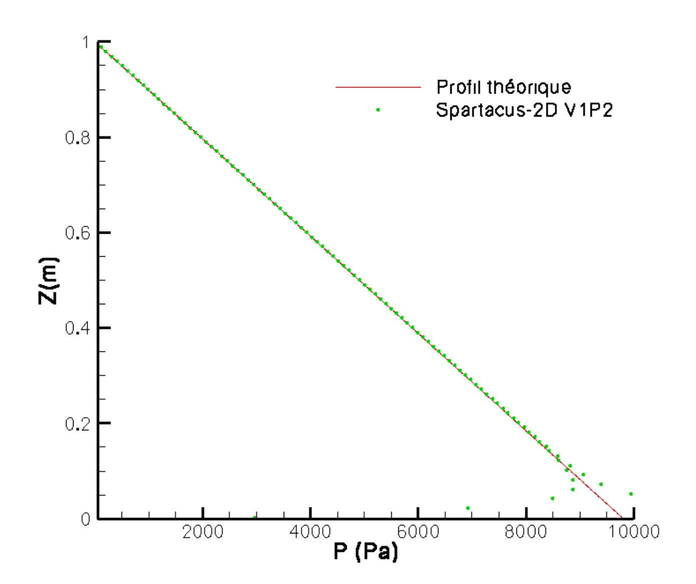


Figure 28 – Distribution verticale de pression à la fin de la phase hydrostatique (cercles) comparée à la distribution théorique hydrostatique (trait).

Le terme de gradient de pression est modélisé selon l'équation (131) (KPRES = 2) et les effets visqueux par la formulation (132) (KVISQ = 1), et deux modèle de turbulence sont successivement examinés afin d'en comparer la valeur (modèle de longueur de mélange et du modèle k–L_m). Le modèle k–ε n'a pas été retenu ici, bien qu'il donne des résultats satisfaisants en première approche (Violeau, 2004). On retient le modèle de production classique (KPROD = 1), le modèle de déformation standard (KDEF = 1) et le modèle de vitesse de frottement fondé sur la loi logarithmique (KUSTAR = 1), ici utilisé en définissant des points fictifs situés à une distance Δ égale à 5,0δr (DESTDR = 5) de la paroi, ceci afin de ne pas prendre en compte de particules fictives lors du calcul de la vitesse aux points fictifs. La longueur de mélange est spécifiée selon la loi donnée par Nezu et Nakagawa (1993) :

$$L_m = \kappa u_* z\sqrt{1-\frac{z}{H}} \tag{167}$$

qui permet de retrouver le profil logarithmique des vitesses (112). Pour les particules de bord, on définit la longueur de mélange à partir de la même formule mais avec $z = \delta$ (définie au **§ 1.1.4.4**), afin de prendre en compte le fait que les particules de bord ne sont pas situées rigoureusement sur le lit.

Le fichier cas pour la phase dynamique (cas du modèle de longueur de mélange) est présenté ci-dessous :

```
/ Fichiers entrees-sorties
/-------------------------
FICHIER FORTRAN = './canal_turb_dyn_v1p2.f'
FICHIER INITIALISATION =
'../RESULTATS/canal_turb_statique_v1p2_suite'
FICHIER SUITE =
'../RESULTATS/canal_turb_dyn_lm_v1p2_suite'
/FICHIER DE MAILLAGE = '../maillage3'
FICHIER DES POSITIONS =
'../RESULTATS/canal_turb_dyn_lm_v1p2_positions'
FICHIER DES RESULTATS =
'../RESULTATS/canal_turb_dyn_lm_v1p2_champs'

/ Parametres de calcul
/---------------------
NOMBRE DE PAS DE TEMPS = 200000
SUITE DE CALCUL =  OUI
PERIODE DE SORTIE LISTING = 20000
PERIODE DE SORTIE DES CHAMPS = 2000
PERIODE DE SORTIE DES POSITIONS = 200000

/ Parametres geometriques
/------------------------
ABSCISSE MINIMALE DU DOMAINE D ETUDE = 1.E-2
ABSCISSE MAXIMALE DU DOMAINE D ETUDE = 0.11
COTE MINIMALE DU DOMAINE D ETUDE = -4.E-2
COTE MAXIMALE DU DOMAINE D ETUDE = 1.5
PERIODICITE SELON X =  OUI

/ Parametres physiques
/---------------------
GRAVITE = OUI
NOMBRE DE FLUIDES = 1
DENSITES DE REFERENCE = 1000.
VITESSE DU SON = 32.
VISCOSITES MOLECULAIRES = 0.000001

/ Turbulence
/-----------
MODELE DE TURBULENCE = 1
MODELE DE DEFORMATION = 1
MODELE DE PRODUCTION = 2
MODELE DE VITESSE DE FROTTEMENT = 2
RUGOSITE EQUIVALENTE = 0.01
RAPPORT DELTA-DR =  0.1
RAPPORT DEST-DR =  5.0

/ Modelisation des parois
/------------------------
PAROIS MOBILES = NON
TYPE DE CONDITION DE PAROI = 1

/ Moteur de l'ecoulement selon x
/-------------------------------
MOTEUR = 3
VALEUR DE LA VITESSE DEBITANTE = 1.0

/ Parametres numeriques
/----------------------
MODELE DE GRADIENT DE PRESSION = 2
MODELE DE VISCOSITE = 1
```

```
TYPE DU NOYAU = 2
RAPPORT H-DR = 1.5

/ Artifices numeriques
/---------------------
AMORTISSEMENT SELON X = NON
/COEFFICIENT D AMORTISSEMENT SELON X = 5000.
AMORTISSEMENT SELON Z = NON
/COEFFICIENT D AMORTISSEMENT SELON Z = 5000.
LISSAGE DE LA DENSITE = NON
/FACTEUR DE LISSAGE = 0.05

/ Visualisation graphique
/------------------------
POST-PROCESSEUR = 2

&ETA
&FIN
```

Aucun fichier source n'est modifié dans le fichier FORTRAN, à l'exception de spartacus2d.f, qui contient les nombres maximaux de particules (voir **§ 2.4.2.4**) cohérents avec les données de la **table 2**, ainsi que initial.f, afin de spécifier le profil de vitesse initial (164), et melange.f pour imposer L_m selon (167). Lors de la phase de dimensionnement de tableaux, le système réserve 104 Mo de RAM. Cependant, seuls 5,5 Mo sont réellement utilisés lors du calcul, qui a nécessité 63 minutes de simulation sur la machine retenue. Compte tenu de la formule (156) donnant reliant le temps CPU aux caractéristiques de l'écoulement, on trouve $C_T = 2{,}0.10^{-4}$ s, ce qui est relativement élevé comparé aux valeurs annoncées au **§ 1.2.3.2**.

3.2.3 Présentation des résultats et validation

Seul un modèle de turbulence permet de retrouver le profil de vitesse donné par (112). Nous nous limitons donc ici à des valeurs de KTURB égales à 1 (modèle de longueur de mélange) et 2 (k–L_m). Nous comparerons les profils de $\overline{u}$, ε et ν_T aux profils théoriques donnés par exemple dans Viollet *et al.* (1998) :

$$\begin{aligned} u^+ &\equiv \frac{\overline{u}}{u_*} = \frac{1}{\kappa}\ln\frac{\xi H}{k_s} + C_* \\ \varepsilon^+ &\equiv \frac{\varepsilon H}{u_*^3} = \frac{1-\xi}{\kappa\xi} \\ \nu_T^+ &\equiv \frac{\nu_T}{u_* H} = \kappa\xi(1-\xi) \end{aligned} \qquad (168)$$

écris ici sous forme adimensionnelle avec $\xi = z / H$. En revanche, l'énergie cinétique turbulente sera comparée au profil empirique donné par Nezu et Nakagawa (1993) :

$$k^+ \equiv \frac{k}{u_*^2} = C_k \exp(-2\xi) \qquad (169)$$

avec $C_k = 4{,}78$. Bien que ce dernier ne soit pas cohérent avec (168) et (76), il est en effet nettement plus conforme aux observations que la valeur constante donnée par (113). Des données issues de DNS effectuées par Moser, Kim et Mansour (1998) ainsi que des résultats obtenus par Warner *et al.* (2005)

seront aussi utilisées.

La **figure 29** montre, dans chacun des cas, les comparaisons entre profils numériques et théoriques / empirique (sous forme adimensionnelle, conformément aux formules (168) et (169)). La viscosité turbulente donnée par le modèle de longueur de mélange n'est pas montrée, car elle présente d'importantes fluctuations dues au calcul du taux de déformation *S* (Issa, 2005). Ce dernier n'affecte cependant pas la viscosité turbulente avec les autres modèles, les irrégularités de *S* étant lissées par le schéma numérique de l'équation résolue pour *k* (Violeau, 2004). Dans tous les cas, le profil de vitesse demeure lisse et très proche du profil théorique. L'évolution générale de l'énergie cinétique turbulente est cohérente avec les données théoriques et numériques, bien que la simulation présente de notables différences avec les profils de référence. Les différences observées sont certainement liées à une estimation des gradients de vitesse peu précise (Issa, 2005), induisant alors des profils de *S* inexacts. En revanche, l'estimation du taux de dissipation par le modèle à une équation est très satisfaisante.

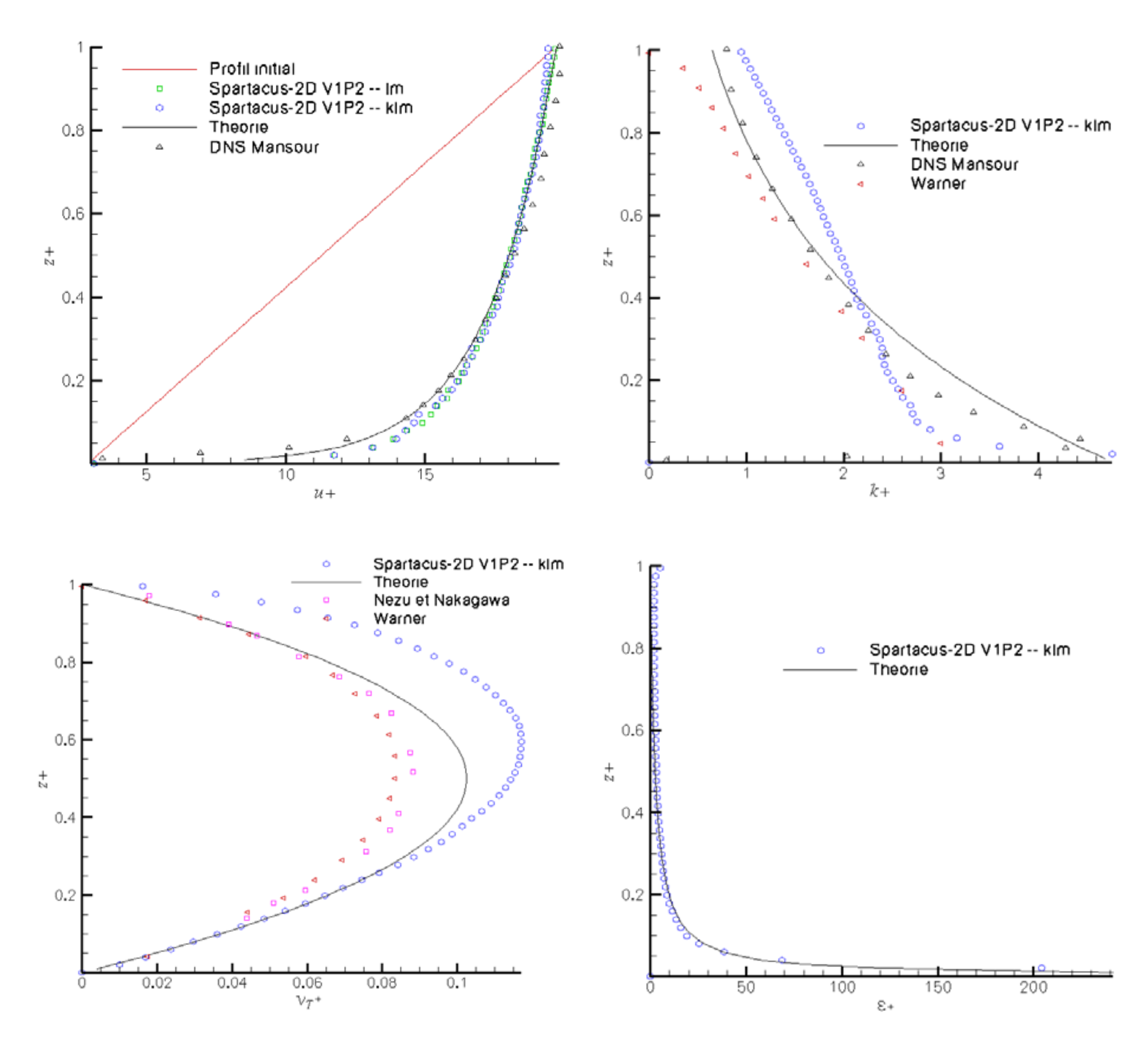


Figure 29 – De gauche à droite puis de haut en bas, distributions verticales de vitesse moyenne, énergie cinétique turbulente, viscosité turbulente et taux de dissipation, adimensionnalisés selon (168) et (169).

3.2.4 Références de la fiche n°1

[1] Issa R. (2005), *Numerical assessment of the Smoothed Particle Hydrodynamics gridless method for incompressible flows and its extension to turbulent flows*, rapport de Thèse de l'Université de Manchester (Royaume-Uni), Department of Mechanical, Aerospace and Manufacturing Engineering.

[2] Moser R.D., Kim J. et Mansour N.N. (1998), Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$, *Physics of Fluids* **11**(4):943-945.

[3] Nezu I. et Nakagawa H. (1993), *Turbulence in open-channel flows*, Monographie de l'IAHR.

[4] Viollet P.-L., Chabard J.-P, Esposito P. et Laurence D. (1998), *Mécanique des fluides appliquée. Ecoulements incompressibles dans les circuits, canaux, rivières, autour de structures et dans l'environnement*, Presses de l'Ecole Nationale des ponts et Chaussées.

[5] Violeau D. (2004), *One and two-equations turbulent closures for Smoothed Particle Hydrodynamics*, Proceedings de la VIe International Conference on Hydroinformatics, Singapour, 87-94.

[6] Warner J.C., Sherwood C.R., Arango H.G. et Signell R.P. (2005), *Performance of four turbulence closure models implemented using a generic lengh scale method*, Ocean Modelling **8**:81–113.

3.3 FICHE N°2 : EFFONDREMENT D'UNE COLONNE D'EAU

- Rédacteur : D. VIOLEAU
- Date de rédaction de la fiche : 30 mai 2006
- Logiciel utilisé : SPARTACUS-2D Version 1.2 (V1P2)
- Pré-processeur utilisé : -
- Post-processeurs utilisé : Rubens© – Version 4.1
  Tecplot© – Version 10.0-3-66
- Machine de calcul : PC Linux – Calibre 4 – 2.8 GHz, 2Go RAM

3.3.1 Description physique

Il s'agit de tester la capacité du code à prédire le comportement d'un écoulement turbulent à surface libre en présence de parois solides sous l'effet de la gravité. On considère une colonne d'eau s'effondrant dans un bassin, que l'on peut assimiler à une rupture de barrage schématique. La modélisation des conditions de parois y est soumise à rude épreuve, l'onde de rupture se propageant avec rapidité vers le mur vertical opposé fermant le bassin. Enfin, ce cas montre la possibilité de traiter des surfaces libres complexes, puisqu'on y observe au retour de l'onde un déferlement suivi d'un intense mélange turbulent. L'effet du lissage de la densité est aussi testé.

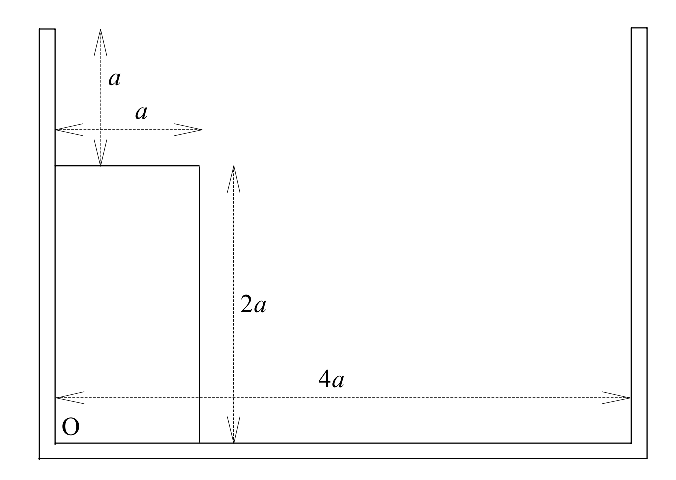


Figure 30 – Géométrie de la colonne d'eau et du bassin.

La **figure 30** donne la géométrie retenue ici, afin de se référer à l'expérience de Koshizuka et Oka (1996). L'origine des coordonnées O est placée au coin inférieur gauche du fluide. La longueur *a* est ici égale à 1 m, ce qui n'a pas vraiment d'importance puisque les validations proposées au **§ 3.3.3** sont adimensionnelles. Le fluide est de l'eau ($\rho_0 = 1000$ kg/m^3, $\nu = 10^{-6}$ m^{2}s^{-1}).

3.3.2 Paramètres numériques

A l'instant initial, les particules sont espacées d'une distance $\delta r = 1$ cm sur une grille régulière et ne

sont animées d'aucune vitesse. Leur densité est égale à la densité de référence et leur pression nulle ; la rapidité de l'écoulement (dominé par les forces d'inertie) permet en effet de s'affranchir d'une phase de repos telle que présentée au **§ 2.4.2.6**.

Un noyau d'ordre 4 (KKERNEL = 2) est ici considéré pour un rapport h / δr = 1,5. Dans un premier temps, les parois sont modélisées par des particules de bord et quatre couches de particules fictives, selon le procédé décrit au **§ 2.4.2.1** (INT(h_t / δr) + 1 = 4). La discrétisation spatiale du système (fluide et parois solides) est décrite dans la **table 2**. Les valeurs de XMIN, ZMIN et XMAX sont choisies selon les données issues de la routine `geometrie.f` (avec une garde de 1 cm), mais ZMAX est augmentée de 50 cm pour minimiser la sortie de particules par le bord supérieur libre (voir fichier des paramètres plus loin). Le fichier `geometrie.f` correspond à l'exemple donné en **Annexe 2.1**.

Particules fluides	20 000
Particules de bord	1 001
Particules fictives	4 044
Nombre total de particules	25 045

Table 2 – Nombre de particules mises en jeu.

La vitesse maximale de l'écoulement est difficile à estimer, mais une première expérience numérique montre qu'elle approche 12 m/s. La vitesse du son c_0 doit donc être au moins égale à 120 m/s, en vertu des recommandations du **§ 1.2.1.4**. La condition portant sur la hauteur d'eau dans (151) ne modifie pas cette valeur, sauf si lors de la montée suivant l'impact sur le mur de droite la hauteur dépasse 15 m, ce qui a peu de chances d'arriver. Il convient en outre d'observer que cette condition repose sur l'hypothèse d'une pression hydrostatique, certainement mise en défaut lors de cette phase de montée compte tenu de l'importante inertie du fluide à ce moment ; on gardera donc c_0 = 120 m/s. Le pas de temps de simulation, fixée ici par l'équation (149), est ainsi de 5.10^{-5} s. Le nombre d'itérations choisi NT = 50 000 permet donc de simuler 2,5 s de temps réel.

Le terme de gradient de pression est modélisé selon l'équation (131) (KPRES = 2) et les effets visqueux par la formulation (132) (KVISQ = 1), et le modèle de turbulence est ici le modèle k–L_m (KTURB = 2) avec une longueur de mélange numérique égale à la distance inter-particulaire (définie par défaut). On adopte l'option de lissage avec un facteur de lissage égal à 0,05 (valeur par défaut).

Le fichier cas est présenté ci-dessous :

```
/ Fichiers entrees-sorties
/-------------------------
FICHIER FORTRAN = './colonne_eau.f'
FICHIER INITIALISATION = '../GEOMETRIE/colonne_eau.init'
FICHIER SUITE = './colonne_eau.suite'
FICHIER DE MAILLAGE = './maillage1'
FICHIER DES POSITIONS = '../RESULTATS/colonne_eau_position'
FICHIER DES RESULTATS = '../RESULTATS/colonne_eau_champs'

/ Parametres de calcul
/---------------------
```

```
NOMBRE DE PAS DE TEMPS = 50000
SUITE DE CALCUL =  NON
PERIODE DE SORTIE LISTING = 100
PERIODE DE SORTIE DES CHAMPS = 1000
PERIODE DE SORTIE DES POSITIONS = 2500

/ Parametres geometriques
/------------------------
ABSCISSE MINIMALE DU DOMAINE D ETUDE = -0.05
ABSCISSE MAXIMALE DU DOMAINE D ETUDE = 4.05
COTE MINIMALE DU DOMAINE D ETUDE = -0.05
COTE MAXIMALE DU DOMAINE D ETUDE = 3.5
PERIODICITE SELON X =  NON

/ Parametres physiques
/---------------------
GRAVITE = OUI
NOMBRE DE FLUIDES = 1
DENSITES DE REFERENCE = 1000.
VITESSE DU SON = 120.
VISCOSITES MOLECULAIRES = 0.000001

/ Turbulence
/-----------
MODELE DE TURBULENCE = 2
MODELE DE DEFORMATION = 1
MODELE DE PRODUCTION = 2
MODELE DE VITESSE DE FROTTEMENT = 2
RUGOSITE EQUIVALENTE = 0.001
RAPPORT DELTA-DR =  0.1

/ Modelisation des parois
/------------------------
PAROIS MOBILES = NON
TYPE DE CONDITION DE PAROI = 1

/ Moteur de l'ecoulement selon x
/-------------------------------
MOTEUR = 0

/ Parametres numeriques
/----------------------
MODELE DE GRADIENT DE PRESSION = 2
MODELE DE VISCOSITE = 1
TYPE DU NOYAU = 2
RAPPORT H-DR = 1.5

/ Artifices numeriques
/---------------------
AMORTISSEMENT SELON X = NON
AMORTISSEMENT SELON Z = NON
LISSAGE DE LA DENSITE = OUI
FACTEUR DE LISSAGE = 0.05

/ Visualisation graphique
/------------------------
POST-PROCESSEUR = 1

&ETA
&FIN
```

Aucun fichier source n'est modifié dans le fichier FORTRAN, à l'exception de spartacus2d.f, qui contient les nombres maximaux de particules (voir **§ 2.4.2.4**) cohérents avec les données de la **table 1**. On y inclut également quelques lignes permettant de déterminer les quantités X et H nécessaires à la validation (**§ 3.3.3**).

La simulation réserve 300 Mo de RAM afin de dimensionner les tableaux utilisés, mais une quantité moindre est réellement utilisée. Elle a nécessité 6 heures 39 minutes de simulation sur la machine retenue. Compte tenu de la formule (156) donnant le temps CPU aux caractéristiques de l'écoulement, on trouve $C_T = 2{,}26.10^{-5}$ s. Rappelons que cette valeur n'est qu'un ordre de grandeur (conforme à ce qui est annoncé au **§ 1.2.3.2**), dépendant de la géométrie du problème considéré, même si les autres cas tests donneront des valeurs semblables.

3.3.3 Présentation des résultats et validation

On s'intéresse à l'abscisse maximale X des particules fluides, ainsi qu'à la hauteur d'eau maximale H le long du mur de gauche (voir **figure 31**). On considère l'évolution dans le temps de ces deux quantités, tracées sous forme adimensionnelle selon la définition :

$$t^* = t\sqrt{\frac{2g}{a}} \qquad X^* = \frac{X}{a} \qquad H^* = \frac{H}{2a} \tag{170}$$

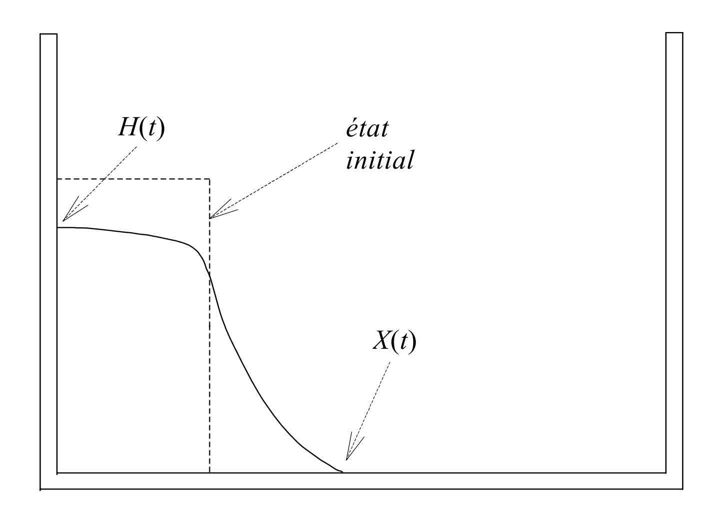


Figure 31 – Définition de *X* et *H*.

La **figure 32** montre les courbes obtenues, comparées au résultats expérimentaux de Koshizuka et Oka (1996). On constate que les résultats sont satisfaisants, bien qu'un léger décalage existe sur X^*, sans doute imputables à la condition à la limite solide près du front, notamment dans les premiers instants de l'écoulement, ou les observations expérimentales fines montrent l'apparition d'un phénomène de déferlement initial encore mal compris (Stansby *et al.*, 1998). Les observations de Koshizuka et Oka montrent en effet, au tout début de la chute de la colonne, une accélération du front plus élevée que ne le suggèrent nos simulations, accélération qui pourrait résulter de ce type de comportement.

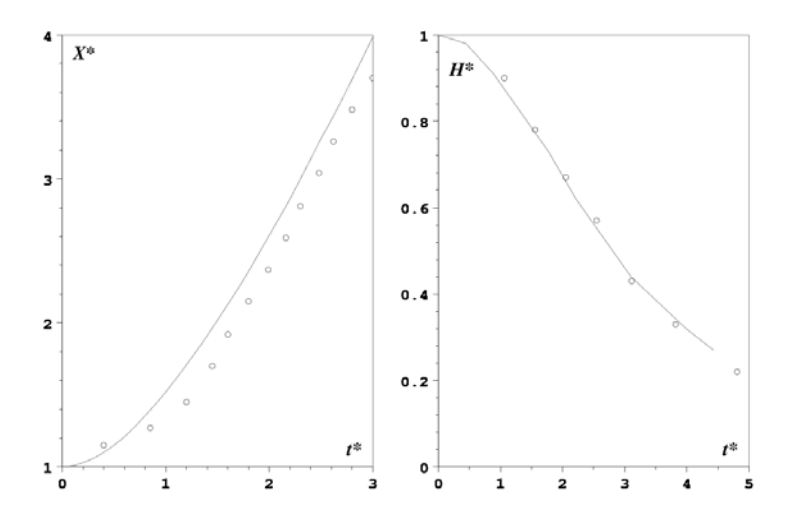


Figure 32 – Evolutions de X^* et H^* en fonction du temps t^* : comparaison des résultats de SPARTACUS-2D V1P2 (traits continus) aux mesures de Koshizuka et Oka (1996) (symboles).

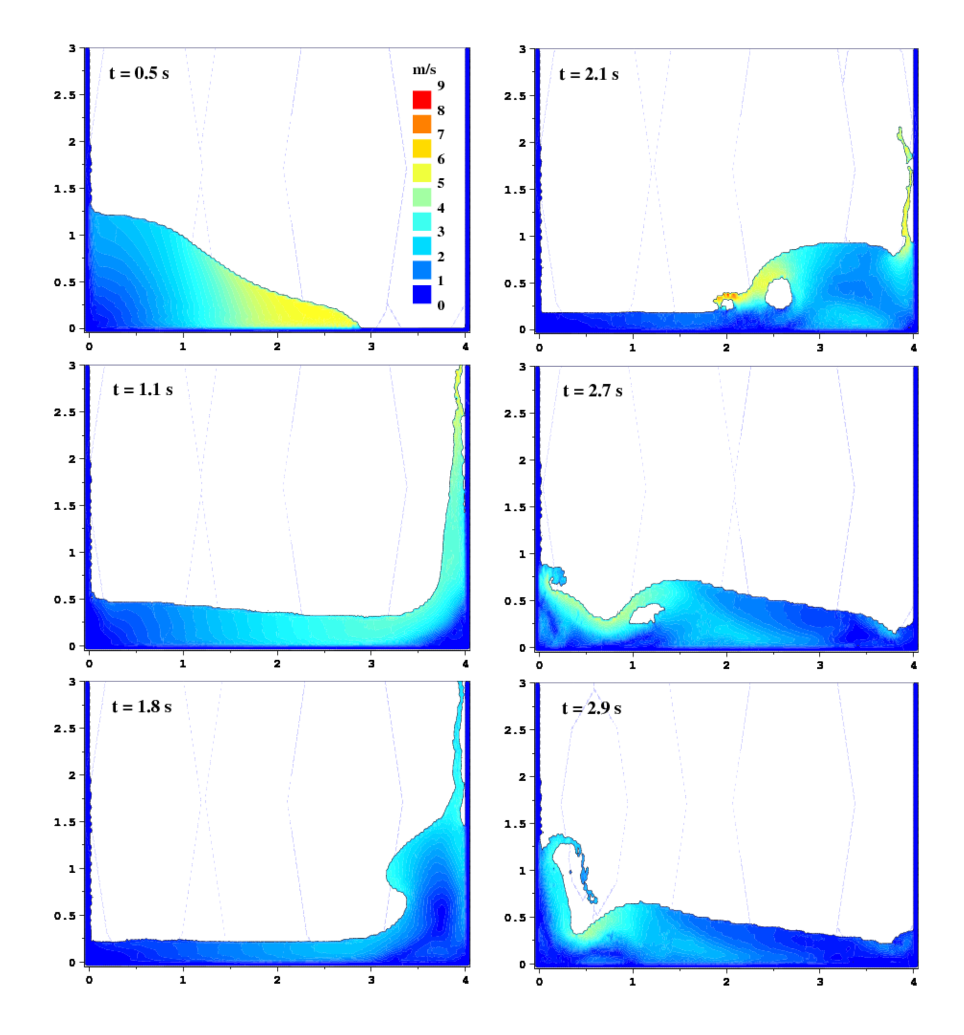


Figure 33 – De gauche à droite et de bas en haut : évolution de l'écoulement lors de la rencontre de la paroi puis du déferlement (les couleurs représentent la norme de la vitesse).

La **figure 33** montre l'évolution de l'écoulement après la rencontre du mur de droite, et l'on voit se produire un déferlement qui ne peut faire l'objet de validations quantitatives.

Afin d'examiner le comportement du modèle de forces de paroi présenté au **§ 1.1.4.2**, nous avons changé le mot-clef `TYPE DE CONDITION DE PAROI` à la valeur `2`, avec `TYPE DE FORCES DE PAROI = 1`, des forces de type Lennard-Jones (`TYPE DE FORCES DE PAROIS = 1`) et un coefficient des forces de paroi égal à $7{,}5g.2a \approx 150$ m^{2}s^{-2} (selon les recommandations du **§ 2.3**). Cet essai oblige à changer la géométrie, les particules fictives devant être retirées afin de ne pas influencer l'écoulement, si bien que le nombre total de particules est réduit de 25 045 à 21 001. La **figure 34** montre que l'allure de l'écoulement est globalement la même, avec cependant des différences locales notoires près des bords. On observe que la méthode des particules fictives est plus stable, bien qu'elle possède le défaut de laisser des particules fluides « adhérer » à la paroi et y demeurer lorsque le fluide proprement dit s'en est éloigné.

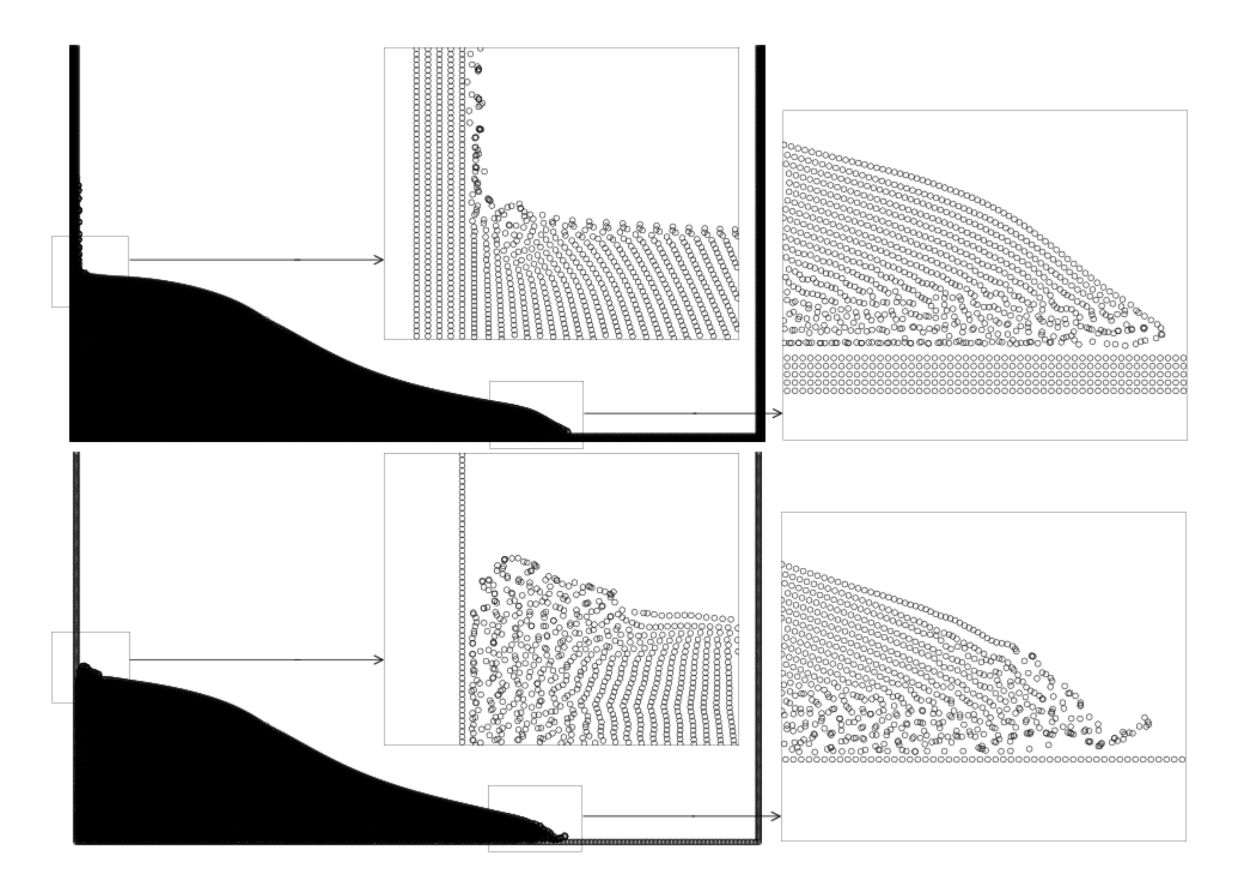


Figure 34 – Position des particules pour *t* = 0,5 s, en haut avec le modèle de particules fictives, en bas avec les forces de paroi.

Enfin, l'influence du modèle de lissage de la densité est illustrée par la **figure 35**, où les graphes de gauche montrent la répartition spatiale de pression à différents instants pour le cas standard déjà cité, tandis que celles de droite montrent ce qu'on obtient en supprimant l'option de lissage. Nous observons que l'effet est extrêmement efficace, surtout lorsqu'on part d'un état hydrostatique résultant d'une phase initiale de repos, testée dans ce seul cas (les images précédentes, rappelons-le, considéraient une phase dynamique immédiate). Cette phase statique a été réalisée dans des conditions

analogues à celle du cas-test de la **fiche n°1**, à l'aide d'une paroi supplémentaire maintenant momentanément le fluide immobile. Remarquons que les graphes de la **figure 35** ont été obtenus avec une vitesse du son égale à 50 m/s au lieu de 120 m/s, car la simulation n'a été poursuivie aussi loin (les vitesses maximales sont largement inférieures à la valeur atteinte lors du choc avec la paroi). Une vitesse du son de 120 m/s permettrait néanmoins de réduire sensiblement les fluctuations de pression mises en évidence, même en l'absence de lissage, mais sans parvenir à la qualité des images de droite de la **figure 35**.

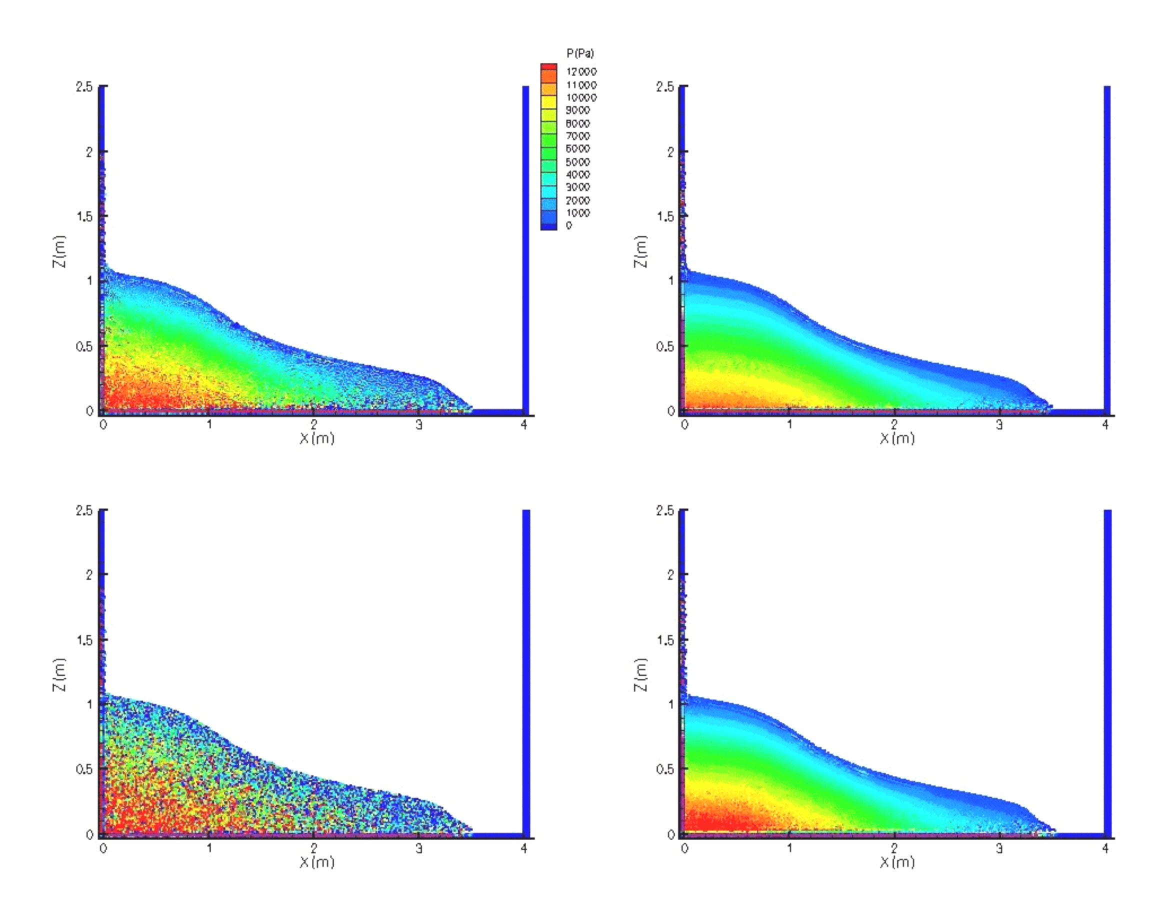


Figure 35 – Répartition spatiale de la pression à *t* = 0,6 s, sans lissage de la densité (à gauche) et avec lissage (à droite). Deux cas sont présentés : en haut, après une phase initiale de repos, à droite en régime dynamique immédiatement.

Ce cas test montre la capacité de SPARTACUS-2D V1P2 à traiter des écoulements gravitaires à surfaces libres très complexes, même en présence de larges efforts. Il montre également la pertinence des modèles de paroi solide et du lissage de la densité.

3.3.4 Références de la fiche n°2

[1] Koshizuka S. et Oka Y. (1996), *Moving-Particle semi-implicit method for fragmentation of compressible fluid*, Nuclear Science Engineering **123**:421–434.

[2] Stansby P.K., Chegini A., Barnes T.C.D. (1998), *The initial stages of dam-break flow*, Journal of Fluid Mechanics **370**:203-220.

3.4 FICHE N°3 : COLLINE PERIODIQUE SANS GRAVITE

- Rédacteur : R. ISSA
- Date de rédaction de la fiche : 13 Février 2006
- Logiciel utilisé : SPARTACUS-2D Version 1.2 (V1P2)
- Pré-processeur utilisé : -
- Post-processeur utilisé : Tecplot© – Version 10.0-3-66
- Machine de calcul : PC Linux – Calibre 4 – 2.8 GHz, 2Go RAM

3.4.1 Description physique

Le principal objectif de ce cas test est d'évaluer l'aptitude du logiciel Spartacus-2D V1P2 à simuler un écoulement confiné sans gravité. On y cherche aussi à prédire une zone de recirculation avec une géométrie périodique. Ce cas-test permet ainsi d'estimer la bonne représentativité des conditions aux limites, tant aux parois solides qu'au niveau des frontières de périodicité. La pertinence du terme de forçage horizontal est aussi évaluée. Une seule restriction importante est faite : on se place en régime laminaire, afin d'éviter les difficultés liées à la définition des conditions aux limites turbulentes en présence de recirculations (gradient adverse de pression, profils de vitesses non logarithmiques, difficultés pour estimer la vitesse de frottement).

On considère un canal à dunes. Grâce à l'application de conditions de périodicité selon la direction x, le système est assimilable à un canal de longueur infinie, caractérisée par la présence périodique de dunes au niveau de la paroi inférieure. La géométrie du système, correspondant à celle définie lors du workshop ERCOFTAC (Uribe et Laurence, 2000), est décrite sur la **figure 36**.

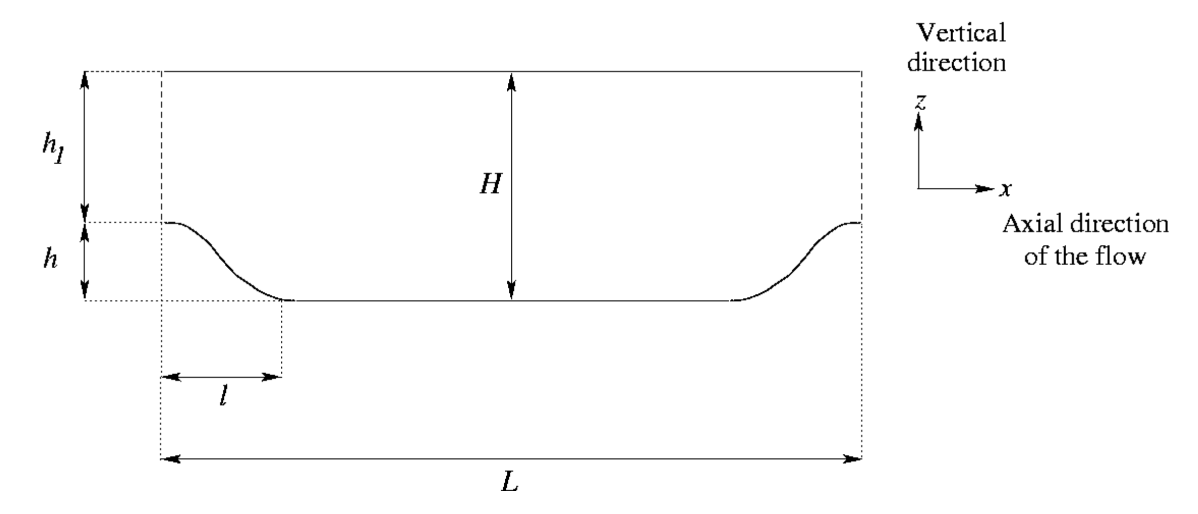


Figure 36 – Géométrie du canal périodique à colline.

La **table 3** indique les valeurs des longueurs retenues. La géométrie exacte de la dune est alors définie par les 6 fonctions suivantes :

$$z(x) = \begin{cases} 28 + 6{,}775.10^{-3} x^2 - 2{,}125.10^{-3} x^3 & si \quad 0 < x < 9 \\ 25.074 + 9{,}755.10^{-1} x - 1{,}016.10^{-1} x^2 + 1{,}889.10^{-3} x^3 & si \quad 9 < x < 14 \\ 25.796 + 8{,}207.10^{-1} x - 9{,}055.10^{-2} x^2 + 1{,}627.10^{-3} x^3 & si \quad 14 < x < 20 \\ 40.464 - x + 1{,}946.10^{-2} x^2 - 2{,}070.10^{-4} x^3 & si \quad 20 < x < 30 \\ 17.925 + 8{,}744.10^{-1} x - 5{,}567.10^{-2} x^2 + 6{,}277.10^{-4} x^3 & si \quad 30 < x < 40 \\ 56.390 - x + 1{,}645.10^{-2} x^2 + 2{,}675.10^{-5} x^3 & si \quad 40 < x < 54 \end{cases} \qquad (171)$$

où x et z désignent respectivement les distances horizontales et verticales, comptées en millimètres à partir de l'origine indiquée sur la **figure 36**.

Grandeur	Valeur (mm)
H	84,98
L	252
l	54
h	28
h_1	56,98

Table 3 – Valeurs numériques des distances indiquées sur la figure 36.

L'écoulement d'eau (ρ_0 = 1000 kg/m^3) considéré ici est laminaire, caractérisé par un nombre de Reynolds Re = 50, ici fondé sur la hauteur h de la colline, la vitesse moyenne débitante U = 1,79.10^{-3} ms^{-1} et la viscosité cinématique ν = 10^{-6} m^{2}s^{-1}.

3.4.2 Paramètres numériques

A l'instant initial, les particules sont espacées d'une distance δr = 1 mm sur une grille régulière et ne sont animées d'aucune vitesse. Leur densité vaut la densité de référence et leur pression est nulle ; l'absence de gravité dispense en effet de simuler une phase de repos comme indiqué au **§ 2.4.2.6**.

Un noyau d'ordre 4 (KKERNEL = 2) est ici considéré pour un rapport h / δr = 1,2. Les parois sont modélisées par des particules de bord et trois couches de particules fictives, selon le procédé décrit au **§ 2.4.2.1** (INT(h_t / δr) + 1 = 3). La discrétisation spatiale du système (fluide et parois solides) est décrite dans la **table 4**.

Particules fluides	19 548
Particules de bord	502
Particules fictives	1 506
Nombre total de particules	21 556

Table 4 – Nombre de particules mises en jeu.

La vitesse du son est égale à 10 fois la vitesse maximale de l'écoulement, conformément aux recommandations du **§ 1.2.1.4** (pas d'effets gravitaires), soit c_0 = 0,03 m/s. Le pas de temps de simulation, fixée ici par la condition de CFL de l'équation (149), est ainsi de $1{,}6.10^{-2}$ s. Le nombre d'itérations choisi NT = 170 000 permet à une particule d'être transportée selon une distance égale à 20 fois la longueur du canal ; le temps physique suffisant à l'établissement d'un régime stationnaire est alors de 2720 s.

Le terme de gradient de pression est modélisé selon l'équation (131) (KPRES = 2) et les effets visqueux par la formulation (132) (KVISQ = 1). Les particules fluides sont mises en mouvement par une force volumique extérieure, ajustée à chaque pas de temps de façon à imposer la vitesse moyenne débitante souhaitée (KMOT = 3 ; voir **§ 2.4.2.3**).

Le fichier cas est présenté ci-dessous :

```
/ Fichiers entrees-sorties
/-------------------------
FICHIER FORTRAN = './colline_lam.f'
FICHIER INITIALISATION = '../GEOMETRIE/colline_lam.init'
FICHIER SUITE =  '../RESULTATS/colline_lam_linux.suite'
FICHIER DES POSITIONS =  '../RESULTATS/colline_lam_position_linux.dat'
FICHIER DES RESULTATS = '../RESULTATS/colline_lam_champs_linux.dat'

/ Parametres de calcul
/---------------------
NOMBRE DE PAS DE TEMPS = 170000
SUITE DE CALCUL =  NON
PERIODE DE SORTIE LISTING =  10000
PERIODE DE SORTIE DES CHAMPS = 10000
PERIODE DE SORTIE DES POSITIONS = 170000

/ Parametres geometriques
/------------------------
ABSCISSE MINIMALE DU DOMAINE D ETUDE  =  0.0
ABSCISSE MAXIMALE DU DOMAINE D ETUDE  = 0.251
COTE MINIMALE DU DOMAINE D ETUDE  =  -4.E-3
COTE MAXIMALE DU DOMAINE D ETUDE  =  8.898E-2
PERIODICITE SELON X =  OUI

/ Parametres physiques
/---------------------
GRAVITE =  NON
NOMBRE DE FLUIDES =  1
DENSITES DE REFERENCE =  1000.
VITESSE DU SON =  0.03
VISCOSITES MOLECULAIRES = 1.E-6

/ Turbulence
/-----------
MODELE DE TURBULENCE =  0
MODELE DE DEFORMATION = 2
MODELE DE PRODUCTION = 2
MODELE DE VITESSE DE FROTTEMENT = 1
RUGOSITE EQUIVALENTE = 0.001
RAPPORT DELTA-DR =  0.1

/ Modelisation des parois
/------------------------
```

```
PAROIS MOBILES = NON
TYPE DE CONDITION DE PAROI =  1
TYPE DE FORCES DE PAROI = 1
COEFFICIENT DES FORCES DE PAROI = 100.

/ Moteur de l'ecoulement selon x
/--------------------------------
MOTEUR =  3
VALEUR DE LA FORCE MOTRICE = 0.
VALEUR DE LA VITESSE DEBITANTE = 1.785E-3

/ Parametres numeriques
/----------------------
MODELE DE GRADIENT DE PRESSION = 2
MODELE DE VISCOSITE = 1
TYPE DU NOYAU = 2
RAPPORT H-DR = 1.2

/ Artifices numeriques
/---------------------
AMORTISSEMENT SELON X =  NON
COEFFICIENT D AMORTISSEMENT SELON X =  0.
AMORTISSEMENT SELON Z =  NON
COEFFICIENT D AMORTISSEMENT SELON Z =  0.
LISSAGE DE LA DENSITE = NON
FACTEUR DE LISSAGE = 0.05

/ Visualisation graphique
/------------------------
POST-PROCESSEUR =  2

&ETA
&FIN
```

Aucun fichier source n'est modifié dans le fichier FORTRAN, à l'exception de spartacus2d.f, qui contient les nombres maximaux de particules (voir **§ 2.4.2.4**) cohérents avec les données de la **table 4**.

A cause de la dimension importante de certains tableaux, la simulation nécessite 300 Mo de RAM. Cependant, seuls 30 Mo sont réellement utilisés lors du calcul. Cette simulation a nécessité 11 heures de simulation sur la machine retenue. Compte tenu de la formule (156) donnant reliant le temps CPU aux caractéristiques de l'écoulement, on trouve $C_T = 2{,}62.10^{-5}$ s, ce qui reste cohérent avec les valeurs déjà annoncées (Cf. autres fiches).

3.4.3 Présentation des résultats et validation

Ne disposant pas de données théoriques ni expérimentales pour ce cas-test, les résultats obtenus avec le logiciel SPARTACUS-2D V1P2 sont comparés à ceux issus du code de calcul eulérien CODE_SATURNE, logiciel développé et validé à EDF R&D / MFEE à partir d'une méthode de volumes finis (Uribe et Laurence, 2000).

La **figure 37** montre que les champs de vitesse axiale et verticale sont très similaires à ceux obtenus avec le CODE_SATURNE. D'autre part, comme l'indique la **figure 38**, la zone de recirculation est clairement simulée par Spartacus-2D V1P2. Cependant, les résultas obtenus ici ne sont pas aussi lisses que ceux relatifs au CODE_SATURNE (notamment en termes de vitesses verticales). Ces irrégularités, en partie imputables au caractère lagrangien de SPARTACUS-2D V1P2, peuvent

néanmoins être considérablement réduites en effectuant un moyennage temporel des résultats, une fois la convergence du système établie (Issa, 2005).

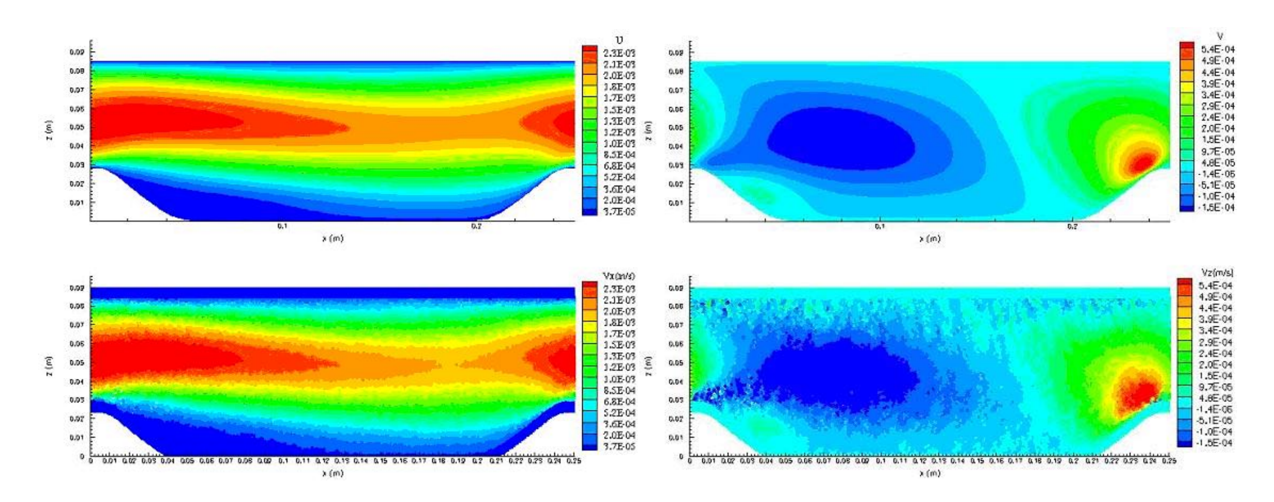


Figure 37 – Comparaison entre le CODE_SATURNE (en haut) et SPARTACUS-2D V1P2 (en bas). Distributions spatiales des vitesses horizontale (à gauche) et verticales (à droite) après convergence.

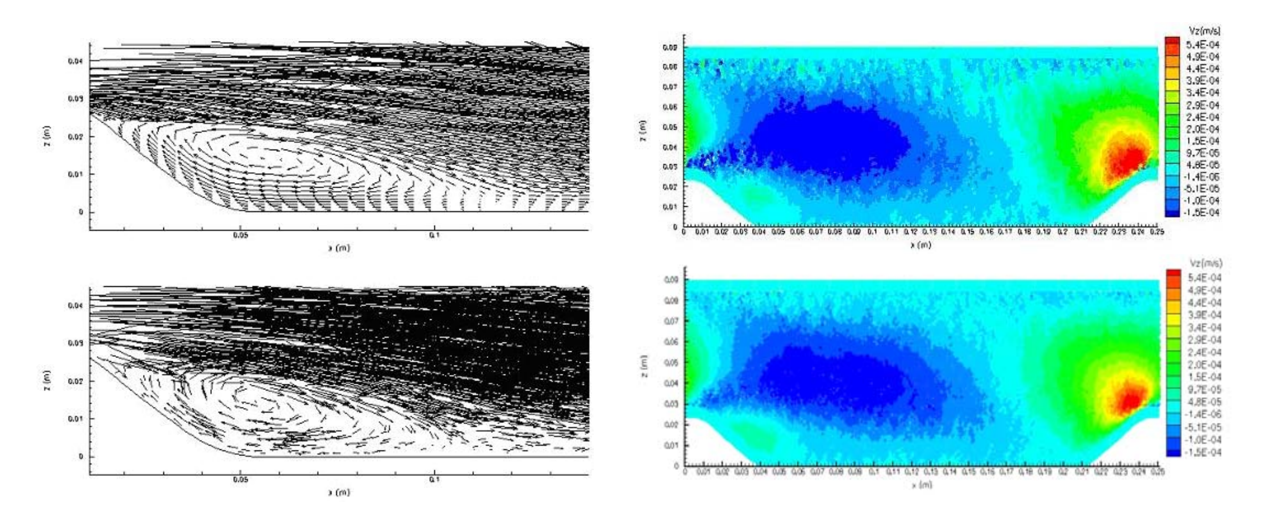


Figure 38 – Comparaison entre le CODE_SATURNE (en haut) et SPARTACUS-2D V1P2 (en bas). Vecteurs vitesse au niveau de la recirculation.

Figure 39 – Distributions spatiales des vitesses verticales avec SPARTACUS-2D V1P2, sans lissage (en haut) et avec lissage de la densité (en bas).

Un autre moyen de réduire ces fluctuations, moins coûteux et directement disponible dans le code, consiste à recourir au modèle de lissage de la densité (mots-clefs `LISSAGE DE LA DENSITE = OUI` et `FACTEUR DE LISSAGE = 0.05`) comme dans le cas-test n°1 (**§ 3.3.3**). Quoique l'effet soit moins convaincant ici, la **figure 39** montre une amélioration sensible.

Les profils de vitesse horizontale (sans lissage de la densité) sont comparés à ceux obtenus avec le CODE_SATURNE le long des coupes P_i (i = 1, …, 6) présentées sur la **figure 40**, d'abscisses respectives $X_1 / h = 0{,}5$, $X_2 / h = 1$, $X_3 / h = 2$, $X_4 / h = 3$, $X_5 / h = 5$ et $X_6 / h = 8$. La **figure 41** montre que les profils obtenus sont très proches. De plus, les profils relatifs aux coupes P_2, P_3 et P_4 révèlent

clairement la présence de la zone de recirculation, dont les caractéristiques sont résumées dans la **table 5**. On constate que le point de détachement est parfaitement simulé par SPARTACUS-2D V1P2. La différence associée au point de réattachement est plus significative et peut être mise en relation avec le caractère lagrangien du code. En effet, le mouvement des particules ne permet qu'une estimation statistique de ce point. Il convient cependant de remarquer que rien ne permet de décider en faveur de l'un ou l'autre des résultats des deux codes en l'absence de données expérimentales. La **figure 42** montre les mêmes profils de vitesse mais en présence du lissage de densité. La prédiction s'en trouve améliorée, notamment la vitesse maximale au niveaux des profils P_2 et P_3. En revanche, le profil P_5 est légèrement moins bon.

	Point de séparation (m)	Point de réattachement (m)
CODE_SATURNE	0,0147	0,120
SPARTACUS-2D V1P2	0,0146	0,115
Différence relative	0,7 %	4,3 %

Table 5 – Prédictions des points de détachement et de réattachement.

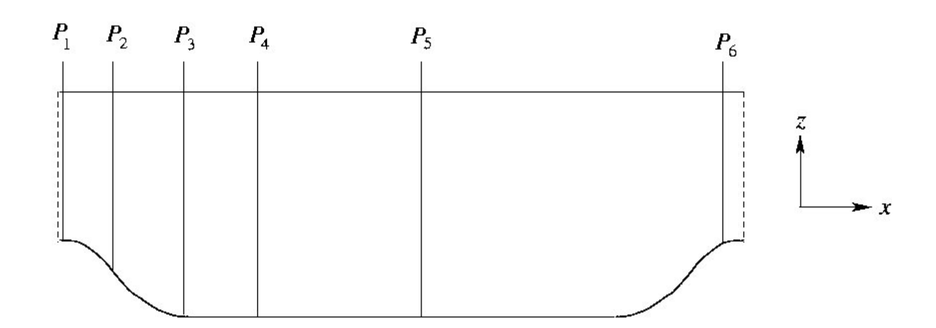


Figure 40 – Localisation des coupes correspondant aux profils de la figure 41.

Ce cas-test atteste donc l'aptitude de SPARTACUS-2D V1P2 à simuler des écoulements incompressibles non gravitaires caractérisés par la présence d'une zone de recirculation en régime périodique. Les profils de vitesse sont en effet très proches de ceux issus d'un modèle eulérien éprouvé. De plus, les conditions aux limites prescrites aux parois et aux frontières périodiques conviennent, ainsi que la modélisation du terme de forçage, puisque le débit voulu est obtenu.

3.4.4 Références de la fiche n°3

[1] Issa R. (2005), *Numerical assessment of the Smoothed Particle Hydrodynamics gridless method for incompressible flows and its extension to turbulent flows*, rapport de Thèse de l'Université de Manchester (Royaume-Uni), Department of Mechanical, Aerospace and Manufacturing Engineering.

[2] Uribe J.C. et Laurence D. (2000), *10th Ercoftac/IAHR Workshop on Refined Turbulence Modelling*, *in* 10th joint ERCOFTAC (SIG-15)/IAHR/QNET-CFD Workshop on Refined Turbulence Modelling.

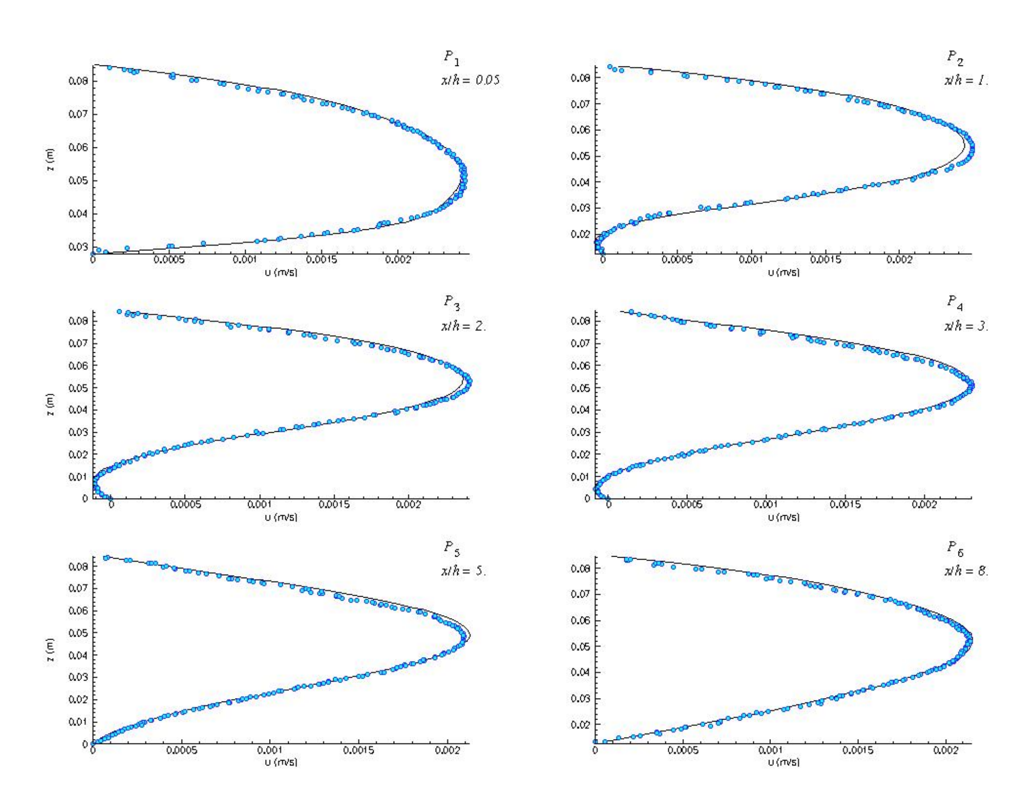


Figure 41 – Comparaison entre le CODE_SATURNE (traits continus) et SPARTACUS-2D V1P2 (cercles). Profils de vitesse axiale pour chacune des coupes indiquées en figure 40, sans lissage de densité.

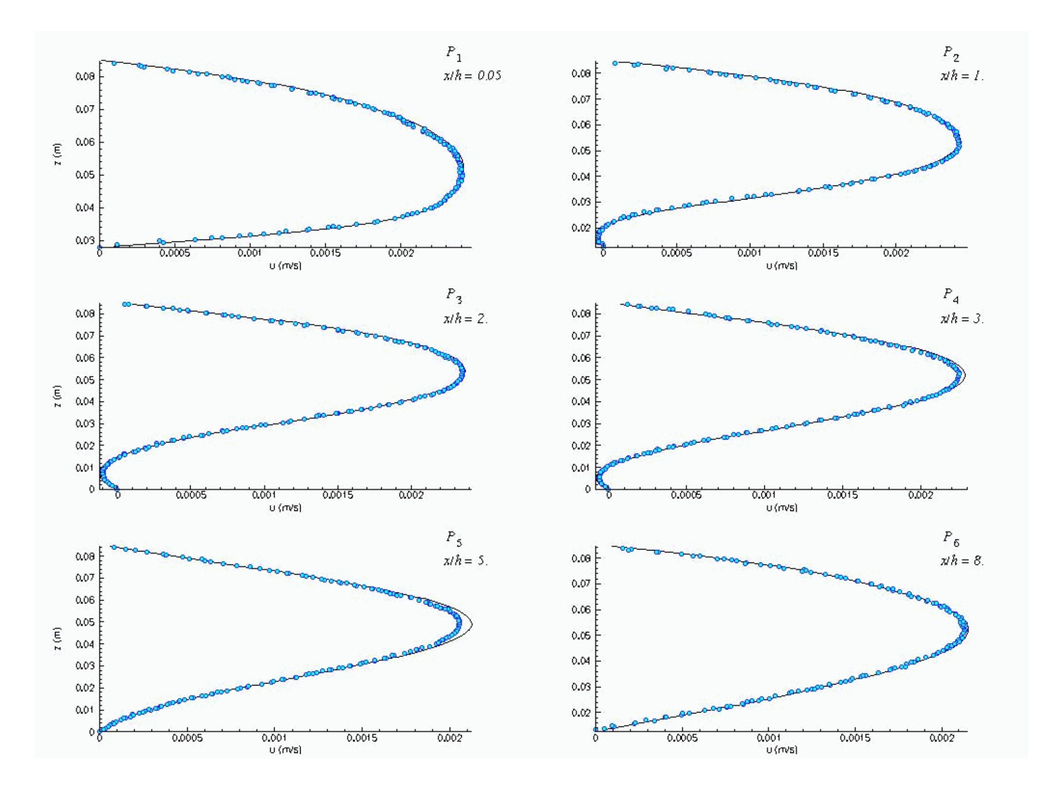


Figure 42 – Comparaison entre le CODE_SATURNE (traits continus) et SPARTACUS-2D V1P2 (cercles). Profils de vitesse axiale pour chacune des coupes indiquées en figure 40, avec lissage de densité.

3.5 FICHE N°4 : ONDE SOLITAIRE

- Rédacteur : R. ISSA
- Date de rédaction de la fiche : 21 Novembre 2006
- Logiciel utilisé : SPARTACUS-2D Version 1.2 (V1P2)
- Pré-processeur utilisé : -
- Post-processeur utilisé : Tecplot© – Version 10.0-3-66
- Machine de calcul : PC Linux – Calibre 4 – 2.8 GHz, 2Go RAM

3.5.1 Description physique

Le premier objectif de ce cas-test est d'évaluer l'aptitude du logiciel Spartacus-2D V1P2 à simuler les différentes étapes du déferlement d'une onde solitaire au niveau d'une plage. Les capacités du logiciel à modéliser correctement la formation de la « seconde vague » (splashup) sont particulièrement examinées. Le second objectif est de montrer l'importance du choix du modèle de turbulence lors de la simulation de tel phénomènes, mettant en jeu de fortes déformations.

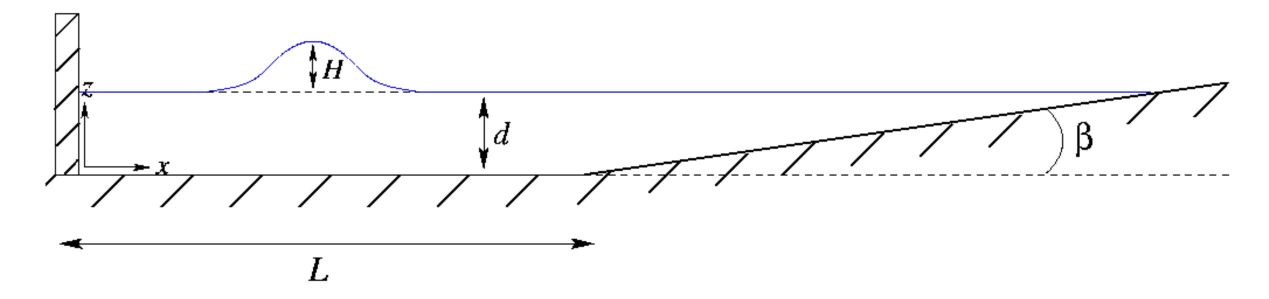


Figure 43 – Géométrie du cas de l'onde solitaire.

La géométrie du système, correspondant à celle utilisée lors de l'expérience décrite par Li et Raichlen (2003), est décrite en **figure 43**. Il s'agit d'un canal présentant une partie plate de longueur L = 5,0 m, terminé par une plage de pente 1/15e et rempli initialement à une hauteur d'eau d = 0,5 m. Il est muni en amont d'un batteur générant une onde solitaire de hauteur initiale H = 0,2 m à l'aide d'un mouvement piloté selon une loi X_0(t) donnée par Hugues (1993) :

$$\dot{X}_o(t) = \frac{\frac{H}{d} CA}{1 + \frac{H}{d} A} \quad (172)$$

avec

$$C = \sqrt{g(d+H)} \qquad A = \frac{1}{\cosh^2\left(K\left(Ct - X_0\right)\right)} \qquad K = \sqrt{\frac{3H}{4d^3}} \quad (173)$$

ce qui donne une relation implicite pour la position du batteur donnée par

$$X_0(t) = \frac{H}{Kd}\tanh\left[K\left(Ct - X_0(t)\right)\right] \tag{174}$$

Afin de générer une onde solitaire de hauteur H, le mouvement a lieu sur une distance S_s définie par

$$S_s = \sqrt{\frac{16Hd}{3}} \tag{175}$$

3.5.2 Paramètres numériques

La discrétisation spatiale du système est donnée dans la **table 4**. Les particules sont initialement espacées d'une distance $\delta r = 2{,}5.10^{-3}$ m selon une grille régulière, et ne sont animées d'aucune vitesse initiale. Un noyau d'ordre 4 (KKERNEL = 2) est ici considéré pour un rapport h / δr = 1,5. Les parois sont modélisées par des particules de bord et trois couches de particules fictives, selon le procédé décrit au **§ 1.1.4.3** (INT(h_t / δr) + 1 = 3).

Particules fluides	334 299
Particules de bord	3 735
Particules fictives	11 205
Nombre total de particules	349 239

Table 4 – Nombre de particules mises en jeu.

La dynamique du système étant relativement rapide, aucune phase statique telle que décrite au **§ 2.4.2.6** n'a été considérée ici, bien qu'on soit en présence d'un écoulement dominé par la gravité. L'écoulement étant turbulent, on se donne un coefficient de rugosité k_s = 0,01 m. La vitesse maximale est ici estimée à partir de la vitesse du batteur, et vaut environ u_{max} = 0,75 m.s^{-1}. La vitesse du son est alors contrainte par la hauteur d'eau (conformément aux recommandations du **§ 1.2.1.4**) et devrait être fixée à 25 m.s^{-1}. Cependant, par sécurité, nous retiendrons ici une valeur c_0 = 50 m.s^{-1},si bien que le pas de temps est de $3{,}0.10^{-5}$ s. Le nombre d'itérations choisi NT = 140 000 permet un déferlement complet de l'onde solitaire (temps réel 4,2 s).

Le terme de gradient de pression est modélisé selon l'équation (131) (KPRES = 2) et les effets visqueux par la formulation (132) (KVISQ = 1). Plusieurs modèles de turbulence ont été testés :

- Viscosité turbulente constante (KTURB = 0). Cette dernière a été fixée à $2{,}5.10^{-4}$ m^{2}s^{-1}, valeur choisie d'après les résultats donnés par les deux autres modèles ;
- Longueur de mélange (KTURB = 1) ;
- Modèle k–L_m (KTURB = 2).

Les tenseurs de déformation et le taux de production sont alors calculés avec les options respectives KDEF = 1 et KPROD = 2.

Le fichier cas ci-dessous correspond à la simulation réalisée avec le modèle k–L_m :

```
/ Exemple de fichier de parametres pour Spartacus2D V1P2
/=========================================================

/ Fichiers entrees-sorties
/-------------------------
FICHIER FORTRAN = vague_deferlante_v1p2.f
FICHIER INITIALISATION =
'../GEOMETRIE/vague_deferlante.init'
FICHIER SUITE =
'../RESULTATS/vague_deferlante_v1p2.suite'
/FICHIER DE MAILLAGE =  '../../..DATA/maillage1'
FICHIER DES POSITIONS =
'../RESULTATS/vague_deferlante_pos_v1p2.dat'
FICHIER DES RESULTATS =
'../RESULTATS/vague_deferlante_champs_v1p2.dat'
TITRE = 'VAGUE DEFERLANTE'

/ Parametres de calcul
/---------------------
NOMBRE DE PAS DE TEMPS = 140000
SUITE DE CALCUL =  NON
PERIODE DE SORTIE LISTING =  1400
PERIODE DE SORTIE DES CHAMPS = 14000
PERIODE DE SORTIE DES POSITIONS = 2800

/ Parametres geometriques
/------------------------
ABSCISSE MINIMALE DU DOMAINE D ETUDE  =  -1.0E-2
ABSCISSE MAXIMALE DU DOMAINE D ETUDE  = 8.375
COTE MINIMALE DU DOMAINE D ETUDE  =  -7.5E-3
COTE MAXIMALE DU DOMAINE D ETUDE  =  0.950
PERIODICITE SELON X =  NON

/ Parametres physiques
/---------------------
GRAVITE =  OUI
NOMBRE DE FLUIDES =  1
DENSITES DE REFERENCE =  1000.
VITESSE DU SON =  50.
VISCOSITES MOLECULAIRES = 0.000001

/ Turbulence
/-----------
MODELE DE TURBULENCE =  2
MODELE DE DEFORMATION = 1
MODELE DE PRODUCTION = 2
MODELE DE VITESSE DE FROTTEMENT = 2
RUGOSITE EQUIVALENTE = 0.001
RAPPORT DELTA-DR =  0.1

/ Modelisation des parois
/------------------------
PAROIS MOBILES = OUI
TYPE DE CONDITION DE PAROI =  1
TYPE DE FORCES DE PAROI = 1
COEFFICIENT DES FORCES DE PAROI = 100.

/ Moteur de l'ecoulement selon x
/--------------------------------
MOTEUR =  0
VALEUR DE LA FORCE MOTRICE = 0.
```

```
VALEUR DE LA VITESSE DEBITANTE = 0.

/ Parametres numeriques
/----------------------
MODELE DE GRADIENT DE PRESSION = 2
MODELE DE VISCOSITE = 1
TYPE DU NOYAU = 2
RAPPORT H-DR = 1.5

/ Artifices numeriques
/---------------------
AMORTISSEMENT SELON X =  NON
COEFFICIENT D AMORTISSEMENT SELON X =  0.
AMORTISSEMENT SELON Z =  NON
COEFFICIENT D AMORTISSEMENT SELON Z =  0.
LISSAGE DE LA DENSITE = NON
FACTEUR DE LISSAGE = 0.

/ Visualisation graphique
/------------------------
POST-PROCESSEUR =  2

&ETA
&FIN
```

La routine paroimobile.f est modifiée comme suit et insérée dans le fichier FORTRAN :

```
C                        ***********************
                          SUBROUTINE PAROIMOBILE
C                        ***********************
C
C DRI
C     .  (KPARMOB, NPMAX, NPART, PI, TEMPS, VXMOB, VZMOB)
     .  (KPARMOB, NPMAX, NPART, PI, TEMPS, VXMOB, VZMOB, X)
C FRI
C
C----------------------------------------------------------------------
C                         MAIN VARIABLES
C .________________.___.______________________________________________
C !            !      !                                              !
C !   NAME     ! MODE !                  MEANING                     !
C !____________!______!______________________________________________!
C !            !      !                                              !
C ! KPARMOB    ! -->  ! MOVING WALL OR EDGE PARTICLE TYPE            !
C ! NPART      ! -->  ! TOTAL PARTICLE NUMBER                        !
C ! NPMAX      ! -->  ! MAXIMUM PARTICLE NUMBER                      !
C ! TEMPS      ! -->  ! PHYSICAL TIME                                !
C ! VXMOB,     ! <--  ! WALL VELOCITY COMPONENTS                     !
C ! VZMOB                                                            !
C !____________!______!______________________________________________!
C
C MODE : -->(NON MODIFIED DATA), <--(RESULT), <-->(MODIFIED DATA)
C----------------------------------------------------------------------
C
C SPARTACUS2D V1P2
C D. Violeau          & R. Issa
C +33(0)1-30-87-78-31 // +33(0)1-30-87-84-28
C LNHE - 2007
C
C FONCTION : impose une vitesse a une ou plusieurs parois
```

```
C FUNCTION : computes the velocity of one or several walls
C
C PROGRAMMES APPELANT : MOUVEMENT
C CALLED BY
C
C PROGRAMMES APPELES  : -
C CALLED PROGRAMS
C
C----------------------------------------------------------------------
C
C Variables
C==========
C
      IMPLICIT NONE
C
      INTEGER I, NPMAX, NPART
C
      DOUBLE PRECISION TEMPS, PERIODE, AMPVIT, PI
C DRI
      DOUBLE PRECISION WH, WD, S, A, C, K
C FRI
      INTEGER KPARMOB (NPMAX)
C
      DOUBLE PRECISION VXMOB (NPMAX), VZMOB (NPMAX)
C DRI
      DOUBLE PRECISION X (NPMAX)
C FRI
      PERIODE = 2.390D0
      AMPVIT  = 0.103D0*2.D0*PI/PERIODE
C DRI
      WD=0.5
      WH=0.2
      C=SQRT(9.81*(WD+WH))
      S=SQRT(4.*WD*WH/3.)
      K=SQRT(3.*WH/(4.*(WD**3.)))
C FRI
      DO 445 I=1,NPART
C
        VXMOB (I) = 0.
        VZMOB (I) = 0.
C
        IF (KPARMOB(I).EQ.1) THEN
C DRI
            IF (X(337850).LE.S) THEN
                 A=1./((cosh(K*((C*(TEMPS))-X(337850))))**2.)
                VXMOB (I) = A*C/(A+(WD/WH))
                VZMOB (I) = 0.
            ELSE
                VXMOB (I) = 0.
                VZMOB (I) = 0.
            ENDIF
C FRI
        ENDIF
C
 445  CONTINUE
C
      RETURN
      END
C
```

A cause de la dimension importante de certains tableaux, la simulation nécessite 2 Go de RAM.

Cependant, seuls 900 Mo sont réellement utilisés lors du calcul. 363 heures ont été nécessaires sur la machine retenue. Compte tenu de la formule (156) reliant le temps CPU aux caractéristiques de l'écoulement, on trouve $C_T = 4{,}98.10^{-5}$ s, ce qui reste cohérent avec les valeurs déjà annoncées (Cf. les autres fiches).

3.5.3 Présentation des résultats et validation

Comme l'indique la **figure 44**, la hauteur *H* de l'onde solitaire générée par Spartacus-2D V1P2 est très proche de la hauteur théorique prescrite à travers les paramètres de mouvement du batteur. Au cours de cette phase de génération de l'onde solitaire, les résultats obtenus avec et sans modèle de turbulence sont identiques car les forces de pression sont alors prédominantes.

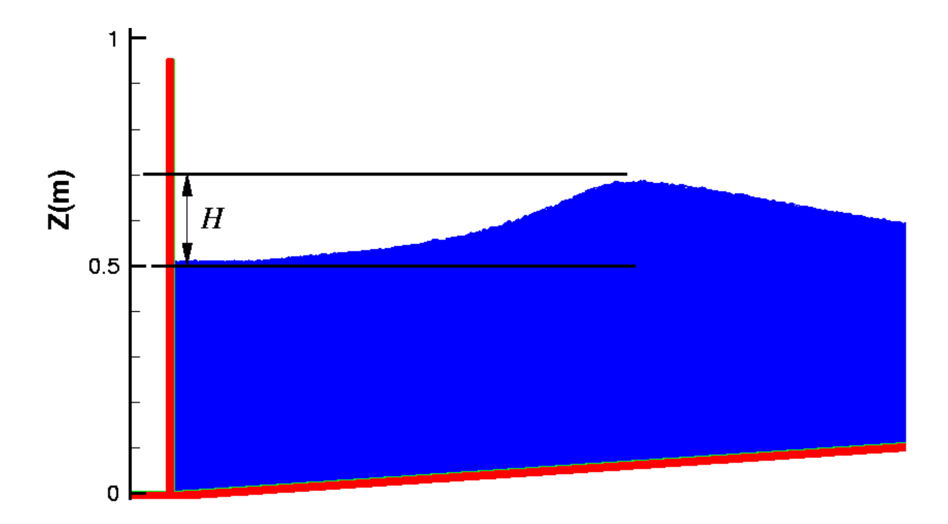


Figure 44 – Génération d'une onde solitaire.

La **figure 45** représente les résultats expérimentaux de Li et Raichlen (2003) à deux instants correspondant au début du déferlement. La forme de la surface libre expérimentale est repérée par une ligne rouge et reportée sur les résultats de simulation afin de permettre une comparaison directe. Les résultats correspondant obtenus avec Spartacus-2D V1P2 montrent tout d'abord l'aptitude du logiciel à reproduire précisément cette étape du déferlement. On constate que les différents modèles de turbulence utilisés (de haut en bas : modèles de viscosité constante, de longueur de mélange et à une équation) ont des comportements très proches, les taux de déformation restant à ces instants relativement modérés. L'écoulement est encore essentiellement piloté par les forces de pression résultant de l'action de la gravité.

Les figures correspondant à l'initiation du phénomène de « splash-up » (**figure 46**) montrent qu'un modèle de viscosité turbulent constant ne reproduit pas correctement ce phénomène tandis que le modèle de longueur de mélange donne des résultats plus satisfaisants. Le modèle à une équation est, quant à lui, très proche de l'expérience. A ce stade, on peut donc affirmer que la modélisation de la turbulence est importante lors de la simulation de tels phénomènes. En effet, la production d'énergie cinétique turbulente y est alors conséquente en raison d'un fort taux de déformation (Hugues, 1993). Les contraintes de Reynolds prennent alors des valeurs relativement élevées, ce qui explique l'importance du choix d'un modèle de turbulence adapté. La forte production d'énergie cinétique

turbulente donne en effet à la viscosité turbulente des valeurs relativement élevées au voisinage du point d'impact, ce que le modèle à viscosité constante ne permet pas de reproduire, si bien que l'écoulement modélisé n'est pas assez visqueux, et la vague déferlante se déplace trop vite par rapport à l'expérience. Le modèle de longueur de mélange permet d'estimer un peu mieux la répartition spatiale de la viscosité (ce qui améliore le résultat obtenu), mais en revanche ne reproduit pas le caractère rapidement variable dans le temps de la distribution de l'énergie cinétique. Le temps caractéristique des grands tourbillons turbulents est en effet alors voisin du temps caractéristique de l'écoulement moyen, ce qui a pour effet de retarder la croissance de la viscosité lors de l'impact. Ce phénomène est pris en compte par le modèle à une équation (équation (136)), ce qui explique l'excellent résultat obtenu.

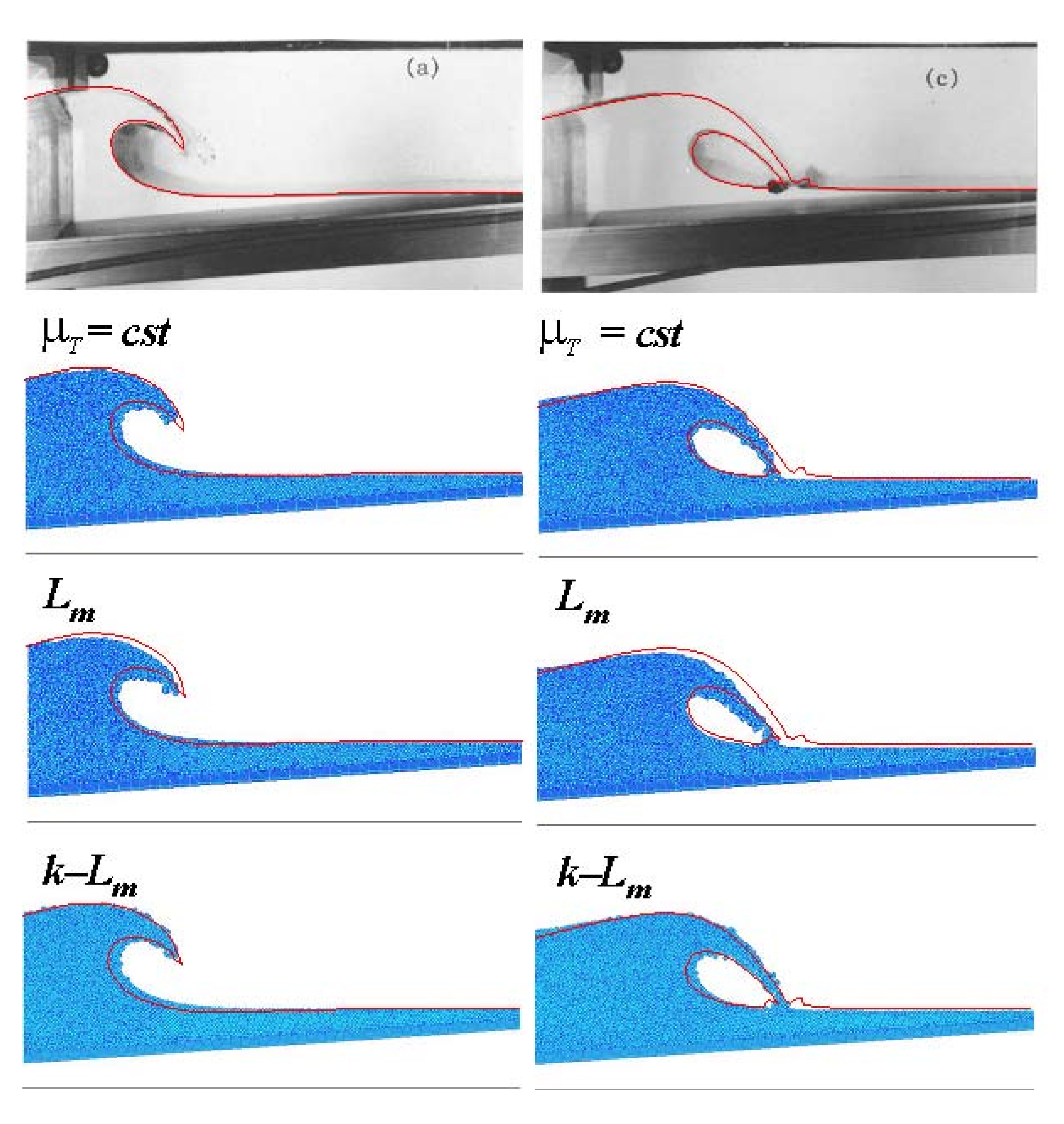


Figure 45 – Déferlement d'une onde solitaire obtenue par l'expérience (figures du haut) et par Spartacus-2D V1P2 (figures du bas) avec différents modèles de turbulence (de haut en bas, viscosité constante, longueur de mélange et modèle à une équation) à deux instants.

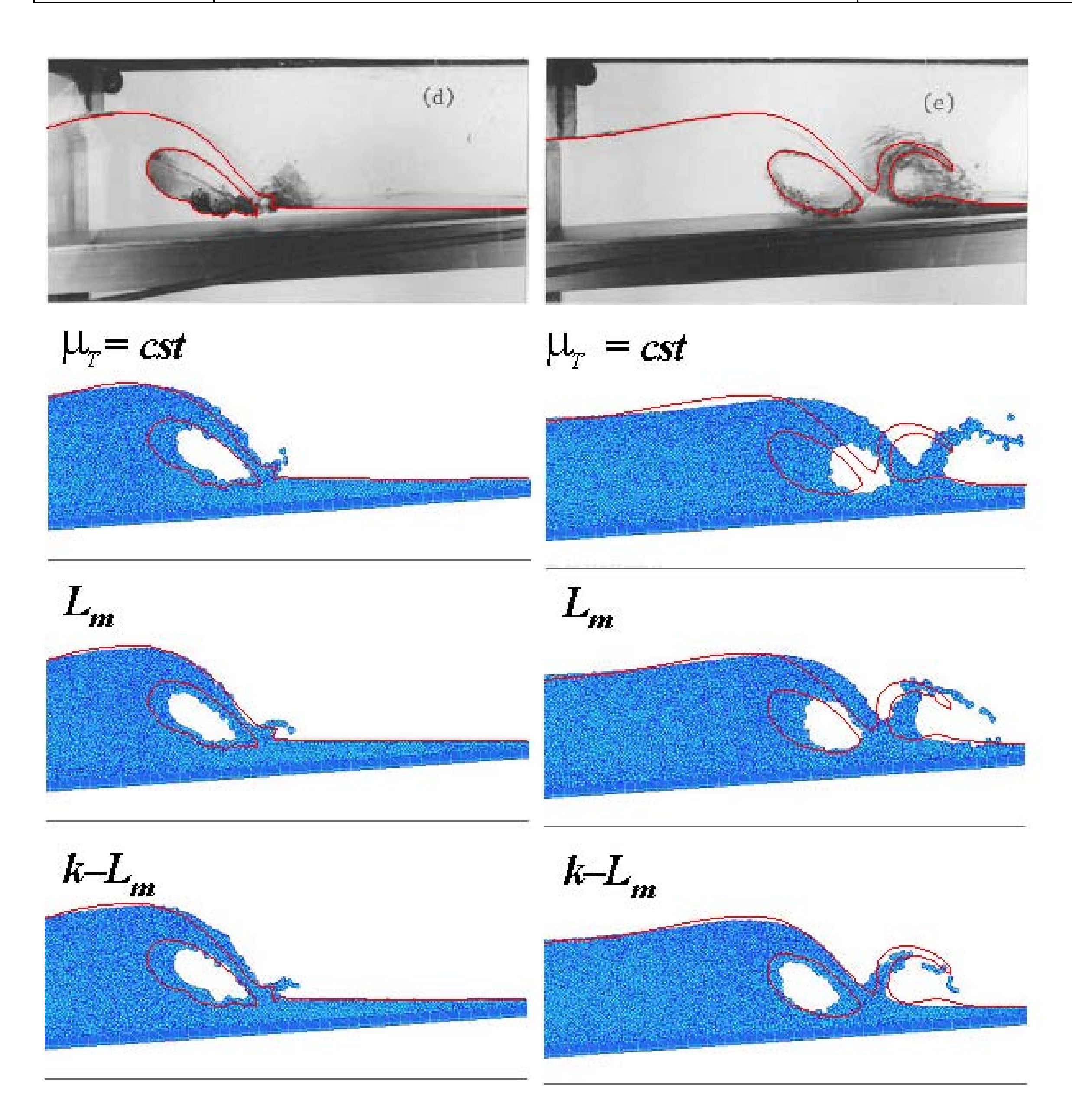


Figure 46 – Même légende qu'en figure 45, à deux instants ultérieurs.

Lors de la phase de développement du « splash-up », la **figure 47** montre cette fois que les trois modèles de turbulence donnent des résultats relativement satisfaisants. Les modèles de viscosité turbulente constante et de longueur de mélange semblent tous deux avoir compensé le « retard » mis en évidence lors de la phase précédente.

La **figure 48** montre une fois de plus qu'un modèle de viscosité constante n'est pas suffisant pour capturer l'onde « de retour » (contre-déferlement) mise en évidence expérimentalement. Les autres modèles semblent légèrement plus adaptés mais présentent encore des déficiences, sans doute en termes de conditions aux limites (notamment en surface libre). D'autre part, pour de tels phénomènes où les taux de déformation et les effets de courbure sont très importants, les limites du modèle de Boussinesq (64) peuvent être atteintes (voir la note du bas de la page 25). Un modèle de viscosité non-linéaire (à l'étude à l'heure où ces lignes sont écrites) pourrait palier ces inconvénients (Violeau et

Issa, 2007). Il faut enfin faire remarquer qu'à mesure que le temps s'écoule, l'expérience est de plus en plus en plus tridimensionnelle, et qu'en outre la forme de la surface libre devient plus difficile à repérer. Les dernières comparaisons demeurent donc forcément très qualitatives.

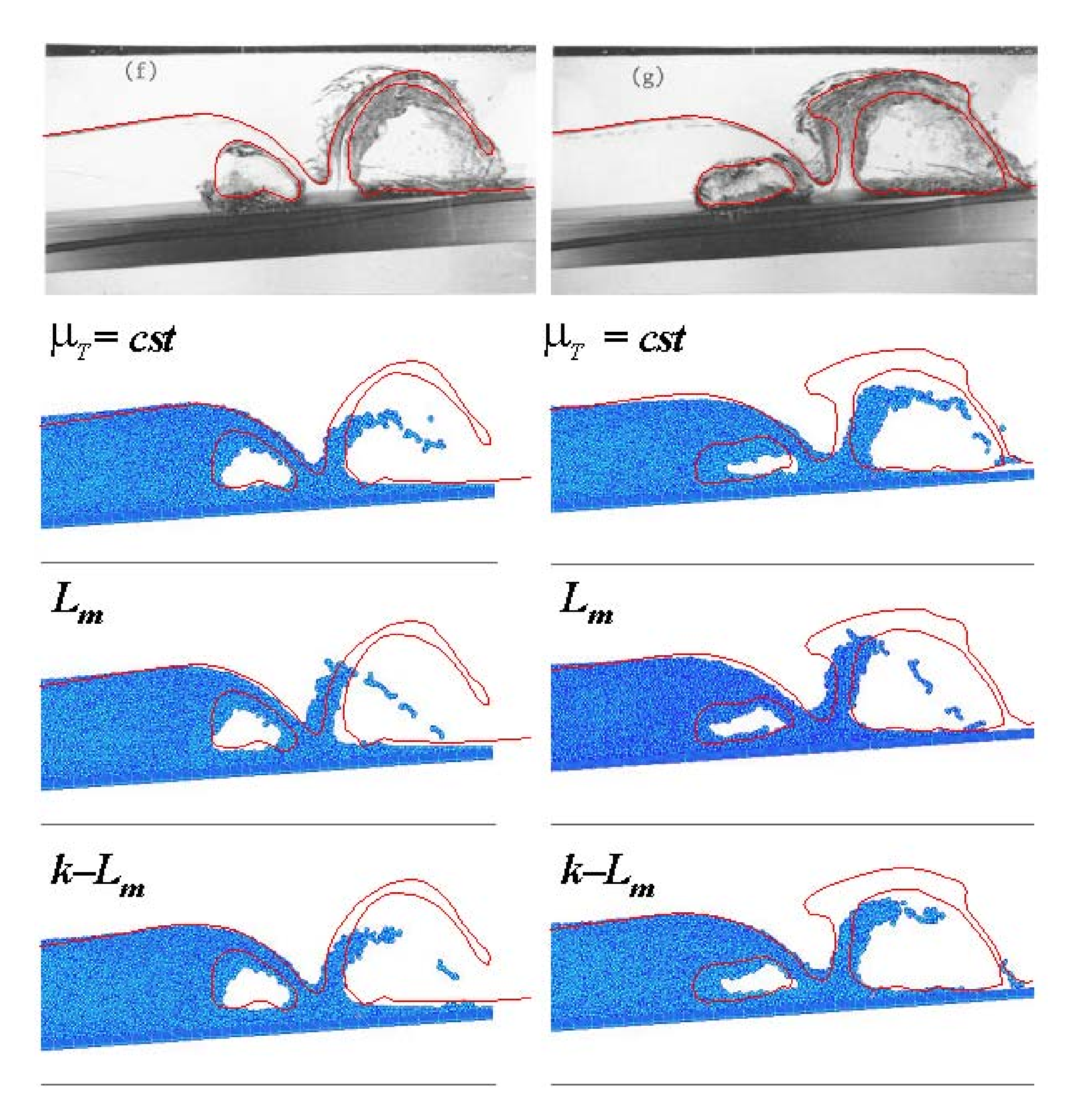


Figure 47 – Même légende qu'en figure 46, à deux instants ultérieurs.

Ce cas-test atteste de l'aptitude de Spartacus-2D V1P2 à reproduire correctement les différentes étapes de déferlement d'une onde solitaire au niveau d'une plage sous réserve que le modèle de turbulence choisi soit suffisamment performant. La modélisation du contre-déferlement reste en revanche délicate dans tous les cas, ce qui montre que des progrès restent à faire dans ce domaine.

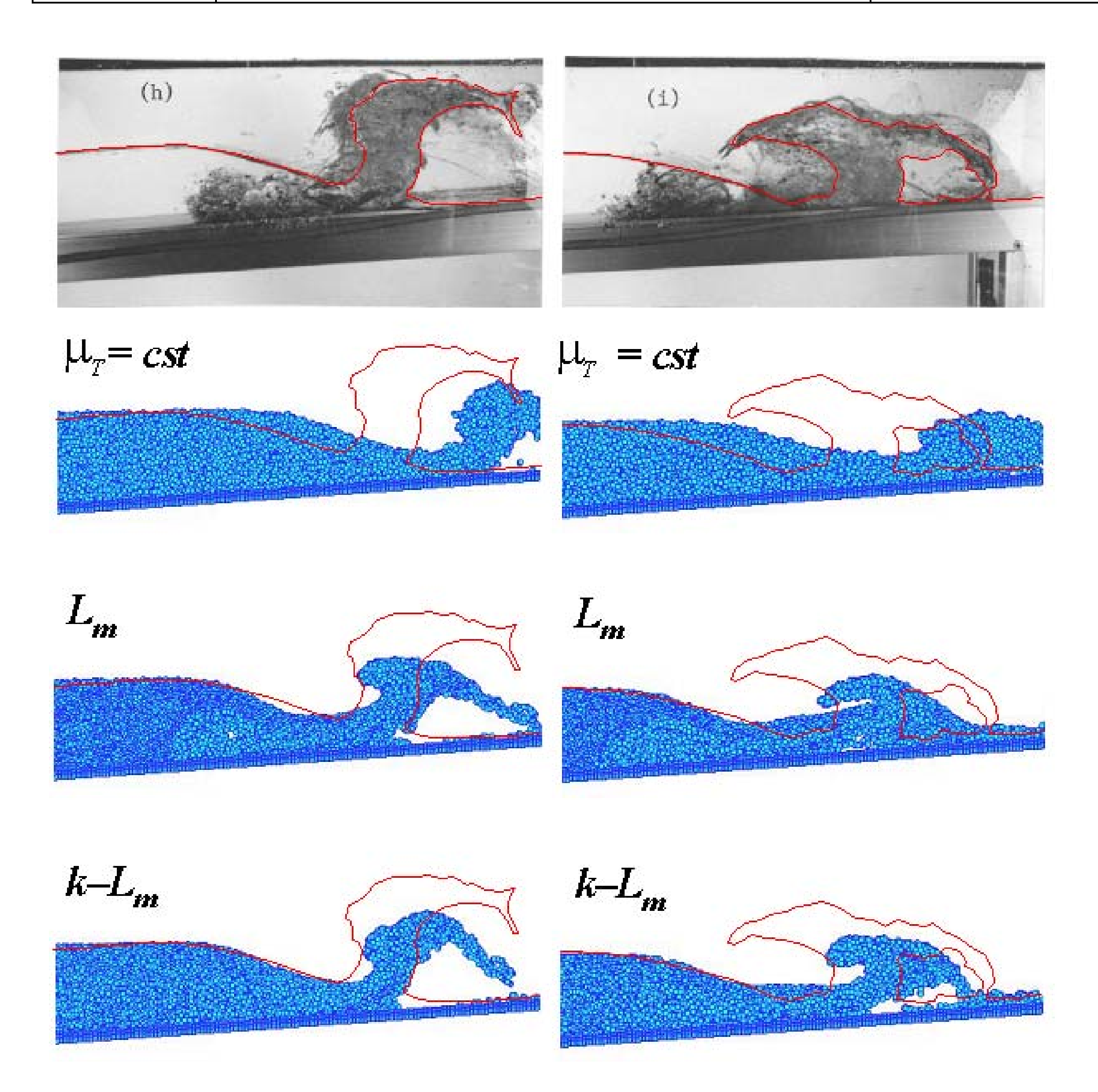


Figure 48 – Même légende qu'en figure 47, à deux instants ultérieurs.

3.5.4 Références de la fiche n°4

[1] Li Y. et Raichlen F. (2003), *Energy balance model for breaking solitary wave runup*, Journal of Fluid Mechanics **456**:295-318.

[2] Hughes S.A. (1993), *Physical models and laboratory techniques in coastal engineering*, World scientific.

[3] Issa R. et Violeau D. (2007), *Modelling a plunging breaking solitary wave with eddy-viscosity turbulent SPH models*, Computers, Materials and Continua (sous presse).

[4] Violeau D. et Issa R. (2007), *Numerical modelling of complex turbulent free surface flows with the SPH gridless method: an overview*, Int. J. Num. Meth. Fluids **53**(2):277-304.

FICHE N°5 : VANNE MOBILE

- Rédacteur : D. VIOLEAU
- Date de rédaction de la fiche : 11 décembre 2006
- Logiciel utilisé : SPARTACUS-2D Version 1.2 (V1P2)
- Pré-processeur utilisé : -
- Post-processeur utilisé : Rubens© – Version 4.1
- Machine de calcul : PC Linux – Calibre 4 – 2.8 GHz, 2Go RAM

3.5.5 Description physique

Le but de ce cas-test est de contrôler l'aptitude de SPARTACUS-2D V1P2 à modéliser correctement une surface libre complexe et rapidement variable, et de comparer les différents modèles de turbulence entre eux d'un point de vue qualitatif.

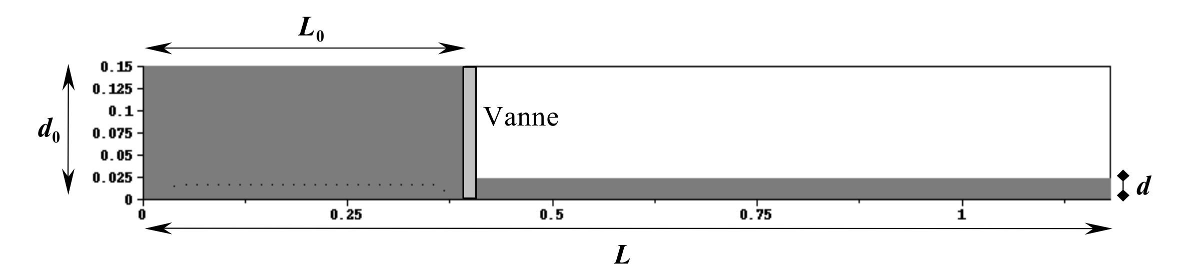


Figure 49 – Géométrie du cas de la vanne mobile.

La géométrie du système, correspondant à celle utilisée lors de l'expérience décrite par Janosi *et al.* (2004), est décrite en **figure 49**. Il s'agit d'un récipient de longueur L = 1,18 m, rempli initialement à une hauteur d'eau d = 1,8 cm ou 3,8 cm. A gauche, une vanne verticale mobile retient à l'instant initial un réservoir d'eau de longueur L_0 = 38 cm et de hauteur d_0 = 15 cm. A l'instant initial, la vanne se relève à une vitesse constante de 1,5 m/s.

3.5.6 Paramètres numériques

La discrétisation spatiale du système est donnée dans la **table 5**. Les particules sont initialement espacées d'une distance δr = 1 mm selon une grille régulière. Un noyau d'ordre 4 (KKERNEL = 2) est ici considéré pour un rapport $h / \delta r$ = 1,5. Les parois sont modélisées par des particules de bord et trois couches de particules fictives, selon le procédé décrit au **§ 1.1.4.3** (INT($h_t / \delta r$) + 1 = 3).

La dynamique du système étant rapide, aucune phase statique telle que décrit au **§ 2.4.2.6** n'a été considérée ici, bien qu'on soit en présence d'un écoulement piloté en partie par la gravité. L'écoulement étant turbulent, on se donne un coefficient de rugosité k_s = 0,2 mm. La vitesse maximale est ici estimée à $u_{\max} \approx \sqrt{gd_0}$ = 1,25 m.s^{-1} ; des essais numériques ont cependant montré qu'elle peut atteindre 2,4 m.s^{-1} à cause d'effets inertiels au moment du déferlement. La vitesse du son est alors fixée à 24 m.s^{-1}, conformément aux recommandations du **§ 1.2.1.4**. Le nombre d'itérations NT =

24 000 couvre un temps réel de 0,60 s nécessaire pour valider le modèle grâce aux expériences de Janosi *et al.* (2004).

Particules fluides	71 310
Particules de bord	1 832
Particules fictives	6 618
Nombre total de particules	79 770

Table 5 – Nombre de particules mises en jeu.

Le terme de gradient de pression est modélisé selon l'équation (131) (KPRES = 2) et les effets visqueux par la formulation (132) (KVISQ = 1). Plusieurs modèles de turbulence ont été testés :

- Longueur de mélange (KTURB = 1) ;
- Modèle k–L_m (KTURB = 2) ;
- Modèle k–ε (KTURB = 3).

Les tenseurs de déformation et le taux de production sont alors calculés avec les options respectives KDEF = 2 et KPROD = 2.

Le fichier cas ci-dessous correspond à la simulation réalisée avec le modèle k–L_m :

```
/ Rupture de barrage en regime turbulent par Spartacus2D V1P2
/============================================================

/ Fichiers entrees-sorties
/-------------------------
FICHIER FORTRAN = './vanne.f'
FICHIER INITIALISATION = '../GEOMETRIE/Vanne.init'
FICHIER SUITE =  './vanne.suite'
FICHIER DE MAILLAGE =  '../../COLONNE_EAU/CAS/maillage1'
FICHIER DES POSITIONS =  '../RESULTATS/vanne.positions'
FICHIER DES RESULTATS = '../RESULTATS/vanne.champs'

/ Parametres de calcul
/---------------------
NOMBRE DE PAS DE TEMPS = 24000
SUITE DE CALCUL = NON
PERIODE DE SORTIE LISTING = 100
PERIODE DE SORTIE DES CHAMPS = 24000
PERIODE DE SORTIE DES POSITIONS = 24000

/ Parametres geometriques
/------------------------
ABSCISSE MINIMALE DU DOMAINE D ETUDE = -0.01
ABSCISSE MAXIMALE DU DOMAINE D ETUDE = 1.191
COTE MINIMALE DU DOMAINE D ETUDE = -0.01
COTE MAXIMALE DU DOMAINE D ETUDE = 0.41
PERIODICITE SELON X = NON

/ Parametres physiques
```

```
/---------------------
GRAVITE = OUI
NOMBRE DE FLUIDES = 1
DENSITES DE REFERENCE = 1000.
VITESSE DU SON = 24
VISCOSITES MOLECULAIRES = 1.E-6

/ Turbulence
/-----------
MODELE DE TURBULENCE = 2
MODELE DE DEFORMATION = 2
MODELE DE PRODUCTION = 2
MODELE DE VITESSE DE FROTTEMENT = 1
RUGOSITE EQUIVALENTE = 0.0002
RAPPORT DELTA-DR = 0.1

/ Modelisation des parois
/------------------------
PAROIS MOBILES = OUI
TYPE DE CONDITION DE PAROI = 1
TYPE DE FORCES DE PAROI = 2
COEFFICIENT DES FORCES DE PAROI = 50.

/ Moteur de l'ecoulement selon x
/-------------------------------
MOTEUR =  0

/ Parametres numeriques
/----------------------
MODELE DE GRADIENT DE PRESSION = 2
MODELE DE VISCOSITE = 1
TYPE DU NOYAU = 2
RAPPORT H-DR = 1.5

/ Artifices numeriques
/---------------------
AMORTISSEMENT SELON X = NON
AMORTISSEMENT SELON Z = NON
LISSAGE DE LA DENSITE = NON
FACTEUR DE LISSAGE = 0.

/ Visualisation graphique
/------------------------
POST-PROCESSEUR = 1

&ETA
&FIN
```

La routine paroimobile.f est modifiée comme suit et insérée dans le fichier FORTRAN :

```
C                         ***********************
                           SUBROUTINE PAROIMOBILE
C                         ***********************
C
     .  (KPARMOB, NPMAX, NPART, PI, TEMPS, VXMOB, VZMOB)
C
C-----------------------------------------------------------------------
C                          MAIN VARIABLES
C .________________.___.________________________________________________
C !           !      !                                                 !
```

```
C !   NAME     ! MODE !                 MEANING                        !
C !____________!______!________________________________________________!
C !            !      !                                                !
C ! KPARMOB    ! -->  ! MOVING WALL OR EDGE PARTICLE TYPE              !
C ! NPART      ! -->  ! TOTAL PARTICLE NUMBER                          !
C ! NPMAX      ! -->  ! MAXIMUM PARTICLE NUMBER                        !
C ! TEMPS      ! -->  ! PHYSICAL TIME                                  !
C ! VXMOB,     ! <--  ! WALL VELOCITY COMPONENTS                       !
C ! VZMOB                                                              !
C !____________!______!________________________________________________!
C
C MODE : -->(NON MODIFIED DATA), <--(RESULT), <-->(MODIFIED DATA)
C-----------------------------------------------------------------------
C
C SPARTACUS2D V1P2
C D. Violeau           & R. Issa
C +33(0)1-30-87-78-31 // +33(0)1-30-87-84-28
C LNHE - 2007
C
C FONCTION : impose une vitesse a une ou plusieurs parois
C FUNCTION : computes the velocity of one or several walls
C
C PROGRAMMES APPELANT : MOUVEMENT
C CALLED BY
C
C PROGRAMMES APPELES  : -
C CALLED PROGRAMS
C
C-----------------------------------------------------------------------
C
C Variables
C==========
C
      IMPLICIT NONE
C
      INTEGER I, NPMAX, NPART
C
      DOUBLE PRECISION TEMPS, PERIODE, AMPVIT, PI
C
      INTEGER KPARMOB (NPMAX)
C
      DOUBLE PRECISION VXMOB (NPMAX), VZMOB (NPMAX)
C
C --- DV debut
c
c      PERIODE = 2.390D0
c      AMPVIT  = 0.103D0*2.D0*PI/PERIODE
c
C --- DV fin
C
      DO 445 I=1,NPART
C
        VXMOB (I) = 0.
        VZMOB (I) = 0.
C
C --- DV debut
c
        IF (KPARMOB(I).EQ.1.AND.TEMPS.LE.0.12D0) THEN
          VXMOB (I) = 0.0D0
          VZMOB (I) = 1.5D0
        ENDIF
```

```
C
C --- DV debut
C
 445  CONTINUE
C
      RETURN
      END
```

La simulation nécessite environ 600 Mo de RAM et 11 heures 30 de temps CPU pour le modèle $k–L_m$. Compte tenu de la formule (156) reliant le temps CPU aux caractéristiques de l'écoulement, on trouve C_T = 7,12.10^{-6} s, ce qui reste cohérent avec les valeurs déjà annoncées (Cf. les autres fiches), bien qu'un peu plus petit. Des essais ont montré que l'influence du choix du modèle de turbulence sur le temps calcul obéit aux valeurs données par la **table 6** (Flament, 2006).

Modèle de turbulence	**Temps CPU** (sans turbulence = 1,00)
Longueur de mélange	1,35
$k–L_m$	1,55
$k–\varepsilon$	1,60

Table 6 – Dépendance du temps CPU vis-à-vis du modèle de turbulence.

3.5.7 Présentation des résultats et validation

L'article de Janosi *et al.* (2004) présente d'assez précises photographie de l'expérience reproduite ici à l'échelle, qui permettent une validation précise en termes de forme de surface libre instationnaire. Les **figures 50** à **53** permettent d'effectuer une comparaison des différents modèles avec l'expérience ; la forme de la surface libre expérimentale est grossièrement reproduite en trait rouge sur les graphiques obtenus à l'aide de SPARTACUS-2D V1P2, avec autant de précision que le permettent l'expérience d'une part (en réalité sensible à de légers effets tridimensionnels), la qualité des photographies d'autre part (voir à ce sujet les commentaires de la fiche n°4, **§ 3.4**). On notera que tous les auteurs ayant travaillé sur ce cas-test ont trouvé un décalage temporel d'environ 0,043 s entre les temps affichés sur les photos et ceux permettant de caler au mieux les résultats, ce qui peut être dû à un choix ambigu de l'instant initial (choisi immédiatement avant ou après le départ de la vanne). Nous adopterons donc cette mesure corrective, ce qui revient à choisir des instant de sorties graphiques situés 0,043 s avant les instants de mesure (ces derniers seront néanmoins retenus comme instants de référence, car ils figurent sur les photographies expérimentales). La forme de la surface libre numérique est déterminée sous Rubens© en traçant des surfaces colorées de densité.

Ces figures montrent que l'allure générale de l'écoulement est très bien reproduite. Des commentaires analogues à ceux de la fiche 4 peuvent être énoncés (**§ 3.4**). En particulier, l'importance du modèle de turbulence se fait sentir dès le déferlement, mais le modèle $k–\varepsilon$ ne semble rien apporter de plus par rapport au modèle à une équation. Le modèle à viscosité constante ne représente pas aussi bien la forme de la surface libre que ces deux derniers, mais étonnement se comporte mieux que le modèle de

longueur de mélange. Le splash-up n'est pas parfaitement reproduit, sans qu'on puisse dire s'il s'agit d'une lacune physique ou numérique.

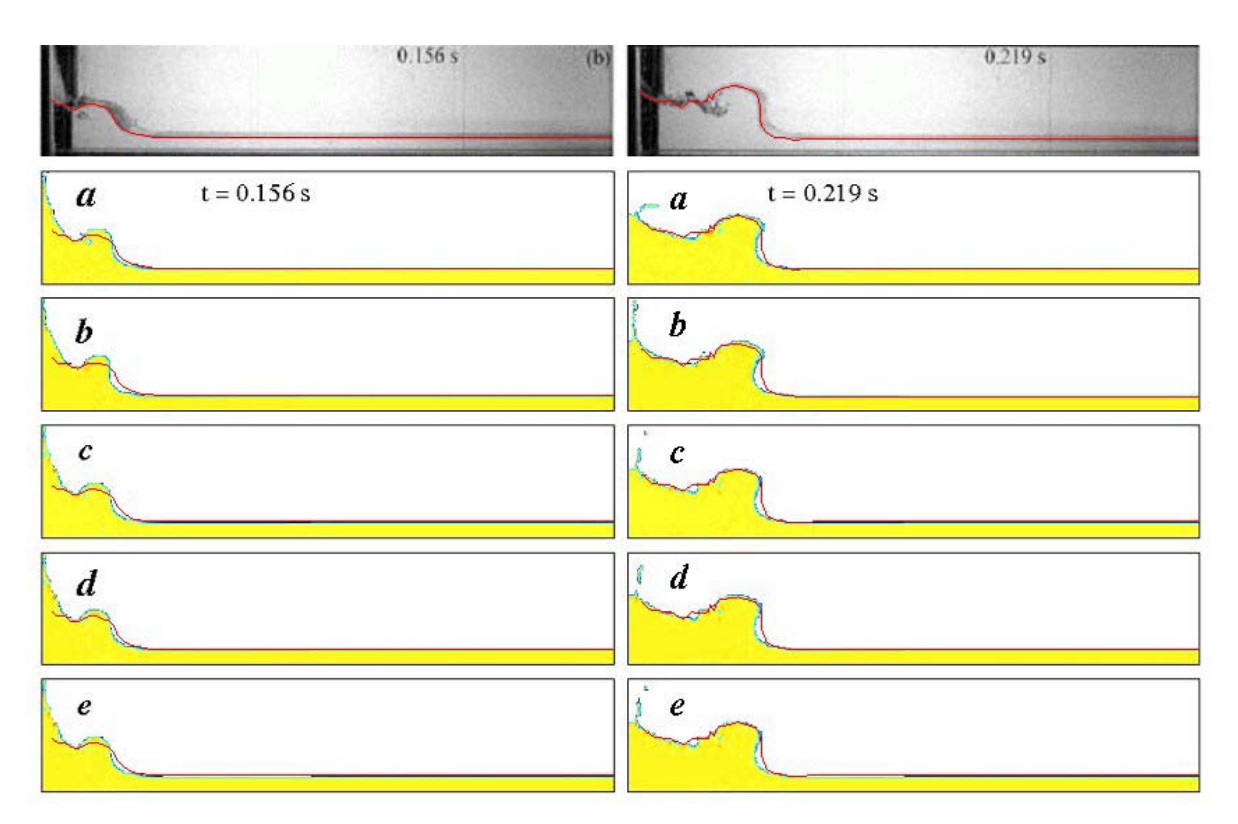


Figure 50 – Cas de Janosi *et al.* (2004) d'après l'expérience (figures du haut) et par Spartacus-2D V1P2 (figures du bas) avec différents modèles de turbulence (de haut en bas, viscosité constante, longueur de mélange, modèle à une équation et modèle *k*–ε) à deux instants.

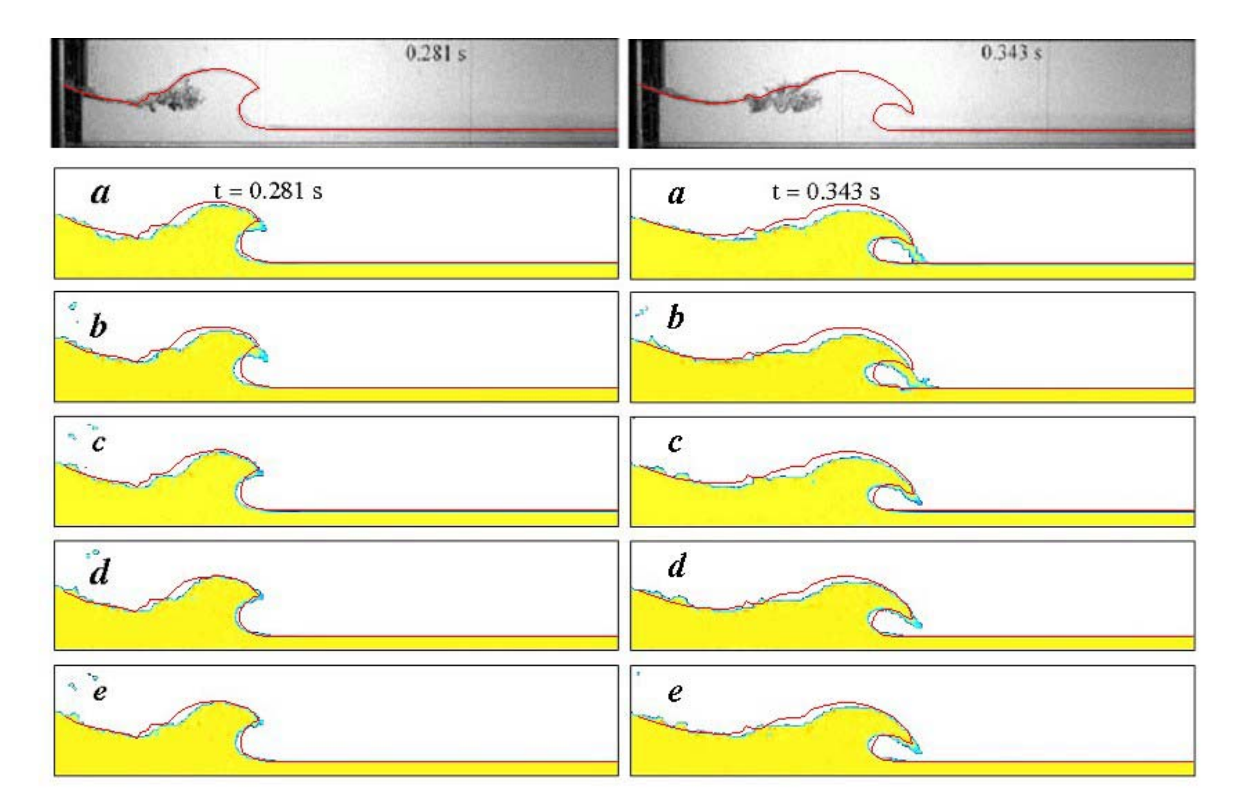


Figure 51 – Même légende qu'à la figure 50, à deux instants ultérieurs.

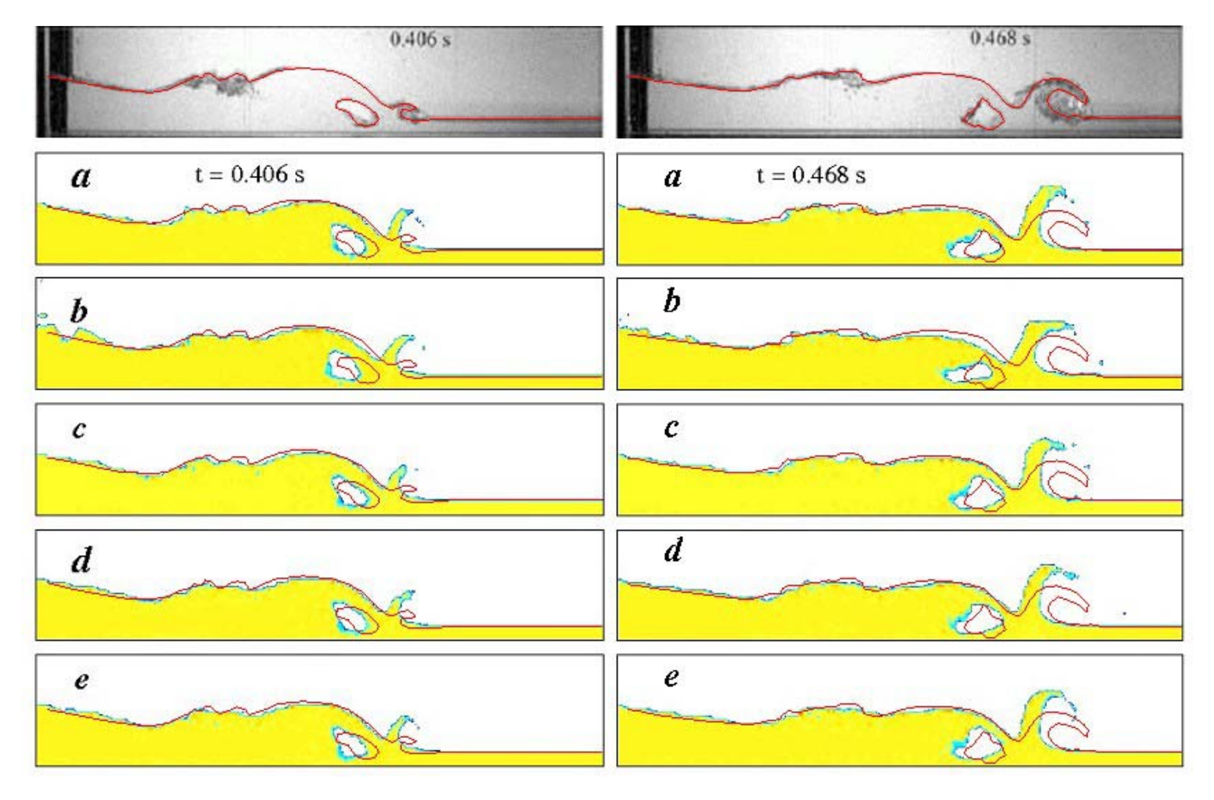


Figure 52 – Même légende qu'à la figure 51, à deux instants ultérieurs.

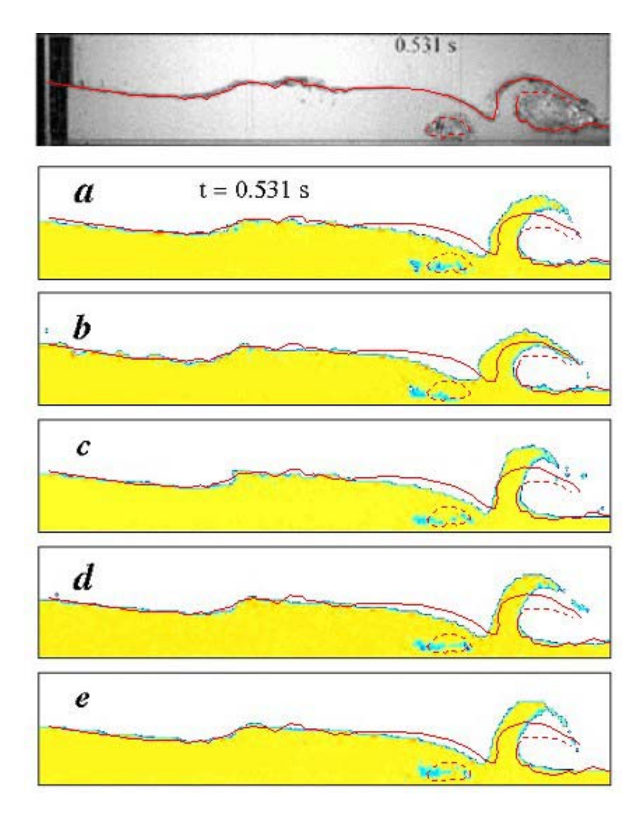


Figure 53 – Même légende qu'à la figure 52, à un instant ultérieur.

3.5.8 Références de la fiche n°5

[1] Janosi I.M., Jan D., Szabo K.G. et Tel, T. (2004), *Turbulent drag reduction in dam-break flows*, Experiments in Fluids **37**:219-229.

[2] Flament H. (2006), *Intégration d'un modèle dans le logiciel lagrangien SPARTACUS-2D*, rapport de stage de DEA de l'Université Pierre et Marie Curie.